

University of Groningen

Monte Carlo simulation of particle interactions at high dynamic range

Ormel, C. W.; Spaans, M.

Published in:
Astrophysical Journal

DOI:
[10.1086/590052](https://doi.org/10.1086/590052)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2008

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Ormel, C. W., & Spaans, M. (2008). Monte Carlo simulation of particle interactions at high dynamic range: Advancing beyond the googol. *Astrophysical Journal*, 684(2), 1291-1309. <https://doi.org/10.1086/590052>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

MONTE CARLO SIMULATION OF PARTICLE INTERACTIONS AT HIGH DYNAMIC RANGE: ADVANCING BEYOND THE GOOGOL

C.W. ORMEL AND M. SPAANS

Kapteyn Astronomical Institute, University of Groningen
 P.O. Box 800, 9700 AV Groningen,
 The Netherlands

Draft version April 28, 2008

ABSTRACT

We present a method which extends Monte Carlo studies to situations that require a large dynamic range in particle number. The underlying idea is that, in order to calculate the collisional evolution of a system, some particle interactions are more important than others and require more resolution, while the behavior of the less important, usually of smaller mass, particles can be considered collectively. In this approximation groups of identical particles, sharing the same mass and structural parameters, operate as one unit. The amount of grouping is determined by the zoom factor – a free parameter that determines on which particles the computational effort is focused. Two methods for choosing the zoom factors are discussed: the ‘equal mass method,’ in which the groups trace the mass density of the distribution, and the ‘distribution method,’ which additionally follows fluctuations in the distribution. Both methods achieve excellent correspondence with analytic solutions to the Smoluchowski coagulation equation. The grouping method is furthermore applied to simulations involving runaway kernels, where the particle interaction rate is a strong function of particle mass, and to situations that include catastrophic fragmentation. For the runaway simulations previous predictions for the decrease of the runaway timescale with the initial number of particles N are reconfirmed, extending N to 10^{160} . Astrophysical applications include modeling of dust coagulation, planetesimal accretion, and the dynamical evolution of stars in large globular clusters. The proposed method is a powerful tool to compute the evolution of any system where the particles interact through discrete events, with the particle properties characterized by structural parameters.

Subject headings: gravitation — instabilities — methods: numerical — planetary systems: formation and protoplanetary disk

1. INTRODUCTION

In the field of natural sciences the dynamical evolution of a population is often determined by interactions that operate at short range, e.g., when the members come into contact, but which do otherwise not influence each other. The dynamical evolution of the system is then regulated by discrete events and the system can be described statistically. The most direct interaction is, of course, the merging of two bodies that come into contact. Examples are numerous. In the Earth’s atmosphere water vapour condenses on cloud droplets (tiny water drops), which merge to form rain drops. In biology, the size distribution of many species of animals can be found by assuming animal groups merge when they make contact. The distribution is then determined by a power-law, which seems to agree with observations of, e.g., schools of tuna fish or herds of African buffaloes (Bonabeau et al. 1999). Similarly, a power-law emerges when bigger organisms are assumed to feed on smaller ones (Camacho & Solé 2001). In astrophysics, too, coagulation processes appear. The transition of the dust component in protoplanetary disks – from (sub)micron grains to planets – covers a huge dynamic range: many tens of orders of magnitude in mass. Due to small molecular forces, micron-sized grains will stick when they meet (Blum 2004). How sticking continues as particles grow to macroscopic proportions (~cm or larger) is still unclear but once the km-size is reached gravity will dominate the accretion process (Lissauer 1993; Goldreich et al. 2004). Another

example where coagulation is of importance are dense stellar systems. Although stellar densities are usually too low for collisions to be likely on a Hubble timescale, it has been shown that mechanisms, e.g., mass-segregation or equipartition, conspire to make stellar collisions feasible (Freitag et al. 2006).

The coagulation process, excluding fragmentation and injection, is described mathematically by the Smoluchowski equation (Smoluchowski 1916), which in the continuous limit reads

$$\frac{\partial f(m)}{\partial t} = -f(m) \int dm' f(m') K(m, m') + \frac{1}{2} \int dm' f(m - m') f(m') K(m - m', m), \quad (1)$$

where the terms on the RHS denote, respectively, the loss and gain terms of particles of mass m . The distribution function, $f(m)$, gives the number density of particles (e.g., dust grains, stars, etc.) of mass m ; i.e.,

$$f(m) \cdot dm = \text{number density of particles in mass-interval } (m, m + dm). \quad (2)$$

Given that the initial conditions of the distribution, i.e., $f(m, t = 0)$ are specified, the coagulation is fully determined by the coagulation kernel, $K(m, m')$, which is the product of the cross section of collision and the relative velocity of the particles, $K(m, m') = \sigma(m, m') \Delta v(m, m')$. It gives the rate coefficient (units: $\text{cm}^3 \text{s}^{-1}$) for the collision between two particles of mass m and m' . For only three distinct cases of

TABLE 1
MOMENTS

kernel	K_{ij}	$M_0(t)$	$M_1(t)$	$M_2(t)$
constant	1	$(1 + \frac{1}{2}t)^{-1}$	1	$1 + t$
sum	$\frac{1}{2}(m_i + m_j)$	$\exp[-\frac{1}{2}t]$	1	$\exp[t]$
product	$m_i m_j$	$1 - \frac{1}{2}t$	1	$(1 - t)^{-1}$

NOTE. — Moment expressions for the classical analytic kernels: M_0 describes the particle density, M_1 the mass density (which is conserved) and M_2 is related to the mass peak, m_p , of the distribution. The initial distribution at $t = 0$ is monodisperse, $f(m, t = 0) = \delta(m - 1)$. The numerical factor of $1/2$ in the sum kernel causes its evolution to coincide with the other kernels at small t . The expressions for the product kernel, $K = m_i m_j$ become invalid after the formation of the gel, at $t = 1$. The mean mass $\langle m \rangle$ and peak mass m_p are related to the moments as $\langle m \rangle = M_1/M_0$ and $m_p = M_2/M_1$.

K does equation (1) have a, closed form, analytic solution (Tanaka & Nakazawa 1994). These are:

- The constant kernel, $K = \text{const.}$;
- The additive (or sum) kernel, $K \propto (m_i + m_j)$;
- The product kernel, $K \propto m_i m_j$;

These three kernels each show their own characteristic behavior. This can be seen by considering the moments of the distribution

$$M_p(t) = \int_0^\infty m^p f(m, t) dm. \quad (3)$$

Multiplying the Smoluchowski equation by m^p and integrating over m yields the moment equation (Leyvraz 2003)

$$\frac{dM_p}{dt} = \frac{1}{2} \int_0^\infty dm dm' K(m, m') f(m, t) f(m', t) \times [(m + m')^p - m^p - m'^p], \quad (4)$$

from which $M_p(t)$ can be found. In Table 1 the zeroth, first and second moments as function of time for monodisperse initial conditions are given for the three classes of kernels. The first moment M_1 , the mass density, is conserved. The zeroth moment, M_0 , describes the (decreasing) particle density; it therefore defines the mean mass of the population, $\langle m \rangle = M_1/M_0$. The second moment, M_2 , is of considerable importance in this paper as it traces the mass distribution on a logarithmic scale, i.e., $m f(m) dm = m^2 f(m) d \log m$. We will therefore define the peak mass as $m_p = M_2(t)/M_1(t)$. For most continuous distributions m_p corresponds to the particle sizes that (together) dominate the total mass of the population.

These three kernels are each representative of a very distinct growth behavior. In the constant kernel growth is orderly: not only do $\langle m \rangle$ and m_p evolve linearly with time, their ratio, which indicates the (relative) spread in the distribution, stays constant as well. This relative narrowness of the distribution means that this kernel can be relatively easily simulated by numerical methods. The situation for the sum kernel, however, is different. Initially, ($t \ll 1$) it evolves similarly as the constant kernel, but after $t \sim 1$ it starts to grow exponentially. The exponential nature of the growth not only holds for $\langle m \rangle$ or m_p but also for the spread in the distribution, $m_p/\langle m \rangle \sim \exp[\frac{1}{2}t]$ (see Fig. 1).

The third class of kernels are the runaway kernels, of which $K_{ij} = m_i m_j$ is its analytic representative. From Table 1 a peculiarity can be noted: the moment expressions do not make

sense after a certain amount of time: the mean density will become negative after $t = 2$, whereas $M_2(t)$ becomes infinite at $t = 1$. The reason for this behavior is that at $t = 1$ a runaway particle of mass $m_R(t)$ is created that separates from the rest of the distribution. In theoretical studies of coagulation this particle is called the ‘gel’ and $m_R(t)$ becomes infinite at precisely $t = 1$. In physical situations, where the total mass in the system is finite, $m_R(t)$ grows continuously with time but nevertheless separates from the distribution. This discontinuity in the distribution is also the reason why the moment expressions become invalid after $t = 1$.

In systems where small number statistics (one runaway particle) become important the assumptions underlying the Smoluchowski equation become invalid. Coagulation is more accurately described by the stochastic coagulation equation which assigns a probability that the system is in a specific state $\{n_k\}(t)$, with $\{n_k\}$ the discrete particle numbers for the mass bins k . The Smoluchowski equation only follows when certain assumptions are made, like $\langle n_i n_j \rangle = \langle n_i \rangle \langle n_j \rangle$, which break down when i and j are statistically correlated (Gillespie 1975). Clearly equation (1) is dangerous when it is used to describe the coagulation process – a process that is inherently discrete.

In many situations of physical interest $K = \sigma \Delta v$ cannot be expressed in the simple forms above where analytic solutions are available. If K can be expressed in terms of a simple mass-scaling behavior of the form $K \propto (\text{mass})^\beta$, then semi-analytic solutions are available (Silk & Takahashi 1979). However, in many physical situations β will not be constant over the whole mass range under consideration. For example, in studies of dust coagulation in protoplanetary disks, relative velocities can be divided into several ‘regimes,’ each with a different velocity behavior (e.g., Ormel & Cuzzi 2007). Another example, is the enhancement of the collisional cross section by gravitational focussing, i.e.,

$$\sigma = \pi R_s^2 \left[1 + \frac{2G(m_i + m_j)}{R_s (\Delta v_{ij})^2} \right], \quad (5)$$

where R_s is the sum of the radii of the bodies and G Newton’s gravitational constant. Equation (5) displays a natural break in the scaling of σ with mass: i.e., $\sigma \propto m^{2/3}$ at low masses vs. $\sigma \propto m^{4/3}$ at high masses. The break happens at the point where the relative velocity of the particles starts to become less than the escape velocity $v_{\text{esc}} = \sqrt{2G(m_i + m_j)/R_s}$ of the biggest body, $\Delta v_{ij}/v_{\text{esc}} < 1$.

So in most situations a numerical approach to the coagulation problem is required. Most directly, equation (1) can be integrated numerically. This entails dividing up the mass-axis into discrete bins and applying equation (1) to all of these bins. However, this is a rather elaborate procedure as, e.g., mass conservation is not intrinsically conserved, and most works have actually avoided to explicitly use the Smoluchowski equation. For example, Spaute et al. (1991) and Inaba et al. (1999) use a (fixed) binning method that assumes the mass inside a bin is distributed as a power-law; coagulation is then an interchange of mass between the various bins. Recently, Brauer et al. (2007) have modified a matrix method, the Podolak algorithm, (see also Kovetz & Olund 1969) which distributes the mass of a coagulation event evenly between its two nearest grid points. Wetherill (1990) and Inaba et al. (1999) use a method where the mass is divided over a discrete number of ‘batches,’ each characterized by a single mass. Interactions with other batches and with themselves cause the batches to grow, i.e., move up the mass

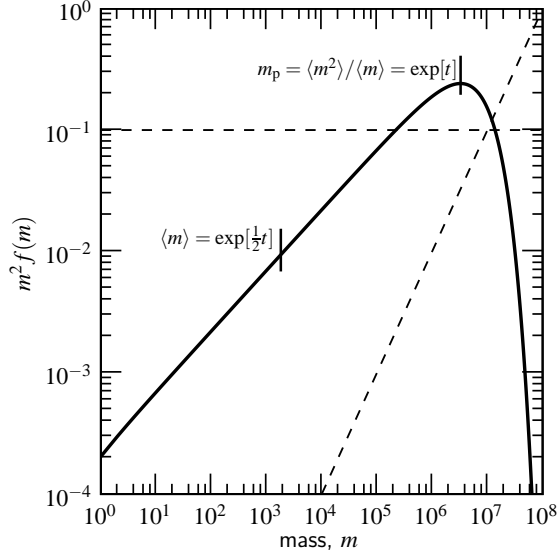


FIG. 1.— The $m^2 f(m)$ mass function of the sum kernel ($K_{ij} = \frac{1}{2}(m_i + m_j)$) at $t = 15$ (solid curve). At this time the mass function has achieved self-similarity. The mean mass, $\langle m \rangle$, and peak mass, m_p , are indicated. Dashed lines of logarithmic slopes 0 and 1, respectively, trace areas of equal mass- and number density per logarithmic unit of mass. Due to the $f(m) \propto m^{-3/2}$ fall-off towards small masses, the small particles dominate by number, yet most of the mass resides in the large particles (around $m = m_p$).

scale. This representation of intrinsically discrete quantities is especially useful in the case of non-continuous distributions (runaway kernels).

Bin size techniques are generally one-dimensional, i.e., the quantities that enter equation (1) (cross sections, velocities) depend on mass only. This approximation may neglect *variables* that play a pivotal role in the coagulation process. For example, a charge distribution on grains (Marshall & Cuzzi 2001; Konopka et al. 2005) is difficult to include in this way. A more subtle example concerns the coagulation of dust particles in disks. Here, collisions affect the internal structure of the particles such that their porosity is not constant, which affects their aerodynamic properties and hence their collisional evolution. Ossenkopf (1993) has extended the Smoluchowski formalism to solve the two-dimensional distribution function for the collisional evolution of dust aggregates, using their fluffiness as the second variable. However, it is clear that the increased number of connections between the bins, caused by the higher dimensionality of the problem, makes the procedure more elaborate, and slower – impractical, perhaps, for three or more dimensions.

In these cases one has to resort to Monte Carlo (MC) simulations. A good description of how a direct-simulation MC method is implemented is outlined by Gillespie (1975) which we will briefly summarize below (§ 2.1). In MC methods $K(m, m')$ determines the probability an event (here: the collision between the particles) takes place. For multi-dimensional models it is probably the preferred method. Besides, the MC method as described by Gillespie (1975) provides an exact description of the stochastic coagulation process, and will also accurately describe runaway kernels. However, in MC simulations one is limited to the finite number of particles that can be followed. It is therefore best suited for distributions that remain relatively narrow, such that a modest number of particles gives a good knowledge of the distribution function f . On the other hand, when the distribution is wide a large number of particles must be present in order to obtain a good characteri-

zation of f ; the required particle numbers then severely limit the applicability of the code.

These effects can best be illustrated in the case of the sum kernel. In Fig. 1 we have plotted the theoretical value of $m^2 f(m)$ at $t = 15$ (Tanaka & Nakazawa 1994) corresponding to $K_{ij} = \frac{1}{2}(m_i + m_j)$, and an initial monodisperse distribution of mass $m_0 = 1$ and $f(m, t = 0) = \delta(m - m_0)$. At $t = 15$ the distribution has reached self-similarity and the shape is the characteristic one for the sum kernel. In Fig. 1 $m^2 f(m)$ is plotted showing (per logarithmic mass interval) the mass density distribution. Due to the $m^{-3/2}$ fall-off of $f(m)$ with mass in the low- m tail of the distribution, the smaller ($m \sim m_0$) particles dominate by number, although it is the larger particles that contain most of the mass. If, for example, the total simulated mass is M_{tot} , there are $\sim M_{\text{tot}}/m_p$ particles representing the mass peak, and $\sim M_{\text{tot}}/\langle m \rangle$ particles in total. The fraction of m_p particles then decreases with increasing m_p as $\langle m \rangle/m_p \propto m_p^{-1/2}$. This behavior inevitably causes computational problems since a few massive particles remain in comparison to a huge number of small particles, while both regimes must be included in order to obtain a good characterization of $f(m)$. With increasing m_p , a point is reached at which there are simply not enough particles to resolve the complete distribution.

In short, the dynamic range for these kinds of simulations is too large to tackle with conventional MC-methods; more particles are required but this is impossible due to computational constraints. To overcome these problems – i.e., to extend MC-methods to situations that require high dynamic range – we will introduce the assumption that one simulation particle can represent multiple physical particles: the grouping method. § 2.2 introduces the grouping method. First the framework of a MC simulation is described in § 2.1, then § 2.2–§ 2.4 outline the grouping method in detail. In § 3 the new method is tested against the analytic solution of the sum kernel (§ 3.1) and the product kernel (§ 3.2). In § 3.3, we consider a class of runaway kernels and test the assumption that the runaway timescale goes down with the size of the simulation (keeping the physical quantities such as the initial mass density constant). Next, in § 3.4 we present steady-state distributions that result from fragmentation of the largest particles. We briefly summarize our findings and discuss astrophysical applications in § 4.

2. THE GROUPING ALGORITHM

Before introducing the grouping method (§ 2.2) we first briefly outline the main features of a direct simulation Monte Carlo code.

2.1. The Monte Carlo code

The direct simulation Monte Carlo method in its simplest form consists of three steps (Gillespie 1975):

- (i) calculation/update of the collision rates, $\{C_{ij}\}$;
- (ii) selection the collision by random numbers: which particles collide and the timestep involved;
- (iii) execution of the collision: removal of the collision partners and addition of the new particle(s).

A Monte Carlo simulation starts with the calculation of the collision rates between the particles, $C_{ij} = K_{ij}/\mathcal{V}$ where \mathcal{V} is the simulated volume and $1 \leq i, j \leq N_p$. Here, N_p is the number of particles in the simulation and C_{ij} , the collision rate, is related to the probability of collision, such that $C_{ij}\Delta t$ is the

TABLE 2
LIST OF SYMBOLS

symbol	description
β	mass dependence of kernels as in $K \propto m^\beta$
Δm_i	mass change of i -particle due to accretion of j -particles
λ_{ij}	collision rate [s^{-1}] between species i and j (including duplicates)
C_{ij}	collision rate [s^{-1}] between particle i and j (excluding duplicates)
K_{ij}	coagulation kernel or rate coefficient [$\text{cm}^3 \text{s}^{-1}$] (velocity \times cross-section)
M_k	k -th moment of distribution
M_{L1}, M_{L2}	first, second most massive particle
N_ϵ	group splitting factor
N_g	number of groups, $\sum_{i=1}^{N_s} w_i$
N_p	number of total particles in simulations
N_s	number of species
N_s^*	target number of species (N_s^*) or groups (N_g^*)
\mathcal{N}	initial number of particles, $N_p(0)$
\mathcal{V}	simulation volume
f_ϵ	maximum allowed fractional mass change of a particle during group collisions
g_i	occupancy number of species i ; number of duplicates
m_\star	characteristic mass of distribution (determining $\{z_i\}$ in equal-mass method)
$\langle m \rangle$	mean mass of the distribution M_1/M_0
m_i	mass of species i
m_p	peak mass of the distribution M_2/M_1
w_i	group number; number of groups for species i
z_i	zoom number of species i

NOTE. — Definitions for key parameters used in this paper.

probability that particles i and j collide in the next infinitesimal time Δt . In total there are $N_p(N_p - 1)/2$ collision rates. A trick which saves memory requirements (although it increases CPU-time) is to store only the partial rates, $C_i = \sum_j^{N_p} C_{ij}$. The total collision rate is $C_{\text{tot}} = \sum_i^{N_p} C_i$. In step (ii) three random deviates (\bar{r}) determine the time interval to the next collision, and the particles involved (i and j). As described by Gillespie (1975) the time increment Δt is found from the total collision rate, $\Delta t = -C_{\text{tot}}^{-1} \ln \bar{r}$, whereas the particles that are involved are found from sampling the $\{C_i\}$ and $\{C_{ij}\}$. Step (iii) computes the new properties of the particles that result from the collision. The fact that in MC-methods these steps are separated shows its advantages when introducing multiple variables; by including multiple variables, the outline of the program, i.e., steps (i)–(iii), will remain the same. Finally, the $(N_p - 1)$ collision rates associated with the newly created particle need to be re-calculated (step (i)), after which the $\{C_i\}$ are updated. The steps then repeat themselves in a new cycle.

2.1.1. Constant- \mathcal{V} vs constant- N simulations

Because each collision that results in coagulation reduces the particle number by one, all particles will have coalesced into one body after N_p steps. This is the constant- \mathcal{V} algorithm: the simulated volume \mathcal{V} and total mass stays the same. However, in many cases this is not the intended behavior: the volume \mathcal{V} occupied by the initial number of particles in the simulation, $\mathcal{N} = N_p(0)$, is only a small sample of the total volume under consideration. Although \mathcal{V} may start out as a volume large enough to represent the particle population, after sufficient coagulation (and decreasing particle numbers) it loses this property: a larger volume is needed in order to get the desired sampling. However, increasing the initial number of particles \mathcal{N} is no solution since the number of collision rates becomes quickly impractical.

A natural solution to prevent the statistical deterioration due to the decreasing particle numbers is to keep the numbers constant: the constant- N algorithm. Here, the number of (simulated) particles is artificially kept constant by *duplicating* a particle after each coagulation event. Thus, mass is added to the system, which physically translates into an expansion of the simulated volume, because the volume-density stays the same. As described in Smith & Matsoukas (1998) the new particle is taken randomly from the existing (simulated) distribution and is therefore called a duplicate. Smith & Matsoukas (1998) have shown that the error introduced due to the duplication scales logarithmically with $\langle m \rangle(t)/m_0$, which is a measure of the amount of growth. This is much improved compared to the constant- \mathcal{V} method in which the reduced number of particles causes the error to scale as the square-root of the growth. In principle, this method allows for indefinite growth. Fundamentally, the assumption is that the simulated volume (containing N_p particles) at all times stays representative of the parent distribution (containing $\gg N_p$ particles). We will test whether the Smith & Matsoukas (1998) duplication mechanism also holds in systems where the growth occurs over many orders of magnitude.

On the other hand, in situations where runaway growth occurs, the collisional evolution will depend on the initial number of particles present in the system, even if locally the same physical conditions apply, i.e., for the same density (Wetherill 1990; Malyshkin & Goodman 2001). Clearly, a constant- N simulation is not the proper way to treat these kernels; we will always use a constant- \mathcal{V} algorithm when treating these kernels. It is challenging to test simulations in which $\mathcal{N} = N_p(0)$ becomes truly astronomical. Here, too, the grouping method will be applied.

2.1.2. The species formalism

A natural way to deal with the existence of a large number of *identical* particles, e.g., resulting from monodisperse initial conditions or through the duplication mechanism in constant- N simulations, is to combine them in the calculation of the collision rates (Spouge 1985; Malyshkin & Goodman 2001; Laurenzi et al. 2002; Laurenzi & Diamond 2003). Since the collision rates of the duplicates are the same, it is efficient to introduce a multiplicity array, g , for all *distinct* particles. This is like saying we have 1 particle of type 1, 5 particles of type 2 ... g_i of species i . It causes the number of distinct simulation particles (N_s , or number of species) to become less than the total number of physical particles in the simulation (N_p) since

$$N_p = \sum_{i=1}^{N_s} g_i. \quad (6)$$

This algorithm is much more efficient, for it is the number of distinct particles (N_s) that matters in the calculation of the collision rates. The collision rate for one collision between species i and j then becomes

$$\lambda_{ij} = g_i g_j C_{ij} \quad (i \neq j); \quad \lambda_{ii} = \frac{1}{2} g_i (g_i - 1) C_{ii} \quad (i = j), \quad (7)$$

where the C_{ij} are the individual collision rates between two single particles. So the λ_{ij} give the probability that a (single) particle of species i collides with one of species j . After the collision partners are determined their multiplicity is first decreased ($g_i \rightarrow g_i - 1$; $g_j \rightarrow g_j - 1$). If this means that g_i becomes zero, no particles of type i exist anymore and it is consequently removed ($N_s \rightarrow N_s - 1$). The new particles that

result out of the collision are added as a new (and distinct) species ($N_s \rightarrow N_s + 1$). When particles are characterized by only one parameter (mass) it would in this case be more efficient to assess first whether the species is really unique, but in this study we treat a general approach. Duplication causes the occupancy number of the duplicated species k to increase by 1, $g_k \rightarrow g_k + 1$. Laurenzi et al. (2002) describe in-depth how the species formalism can be implemented and quantify its computational efficiency.

We note that this grouping of the collision rates is nothing more than a computational trick, which, within the assumptions of, e.g., the duplication mechanism does not approximate the MC-method. Occupation numbers may become very large, though. For example, for monodisperse initial conditions $g_1 = N_p$ initially but g_1 may increase if the constant- N algorithm is enforced and V grows, as in the case of the sum kernel (Fig. 1). In fact, the existence of a large number of identical particles is the basis of the (physical) grouping algorithm.

2.1.3. Limiting N_s .

The accuracy and efficiency of Monte Carlo simulations is then determined by a single parameter, N_s . Ideally, one wants to constrain N_s . In the constant- N_s simulations this means that N_s should fluctuate around a (fixed) target value N_s^* . (There is no need to enforce an exact 1-1 correspondence.) Here we list a few mechanisms that can be applied to balance N_s . Operations that increase N_s :

- Duplication ($g_k \rightarrow g_k + 1$). Although duplication does not directly increase the number of species, the increased multiplicity of a species will decrease the likelihood that it becomes empty of particles ($g_k = 0$) and that it is removed;
- Coagulation/Fragmentation. Because we generally assume that the collision products are different species, the number of species always increases with coagulation and fragmentation. Especially fragmentation may lead to a proliferation of the number of species.

And operations that decrease the N_s parameter:

- (Physical) grouping (discussed below). This has the effect that many particles are involved in the collision and therefore increases the likelihood of a disappearing species ($g_i \rightarrow 0$).
- Coagulation. Requires two particles, with which again a species may disappear.
- Particle removal. This is the reverse of duplication. If the number of species keeps increasing, we may randomly remove particles to enhance the probability of obtaining $g_i = 0$.
- Merging. This is the most drastic approach, in which different species are merged together in a single species whose properties reflect the properties of its progenitors. Because of the averaging, this procedure is somewhat against the spirit of the MC-methods. It is in fact a smoothening where particles of different properties are averaged, including their structural parameters. However, the mean parameters may still be applicable to describe the present state.

In this work the species-averaging is performed as follows. If a species needs to be merged, we first look for its neighbor

species in terms of mass, and then mass-average over their (structural) properties. As in this work mass is the only property, it simply means that for species i and j their merged occupation number becomes $g = g_i + g_j$ and the new mass $(g_i m_i + g_j m_j)/(g_i + g_j)$. (If needed, the latter quantity is rounded to integer values). If structural parameters would be present, however, this procedure can be simply extended to any structural property ψ , i.e., $\psi \rightarrow (g_i \psi_i + g_j \psi_j)/(g_i + g_j)$, where ψ may, e.g., represent charges or filling factor.

2.2. The grouping algorithm

To simulate a coagulation process by Monte Carlo methods in a way to obtain growth over many orders of magnitude, two criteria have to be met:

- I The simulation must involve a large number of physical particles (N_p), such that the population is a proper sample of the continuous distribution. Physically, it means the spatial volume that is sampled will be large enough to account for all various kinds of particles.
- II The number of species (N_s) cannot become too large: since there are $\sim N_s^2$ collision rates to be taken care of, the computational speed will slow down rapidly with increasing N_s .

To meet these conditions – high N_p and low N_s – the occupation numbers g_i must be very large. This can, in principle, be achieved by duplicating particles many times. However, a new problem then arises. Since the total collision rate, C_{tot} , scales with the total number of physical particles squared, i.e., $C_{\text{tot}} \sim N_p^2$, the time step per collision is proportional to N_p^{-2} . The simulations then seem to freeze as the required CPU-time per unit simulation time blows up. In the MC-simulation the focus lies on the particles that dominate by number, which may sometimes be irrelevant from a physical point of view. This occurs, for example, with the sum kernel. Here, the most dominant collisional process – a collision between a small particle and a very massive one – is rather meaningless: a single collision neither affects the mass of the massive particle nor the number of small particles. The ubiquity of many of these subleading collisions then overwhelms the more important ones involving more massive particles, a process that together is very CPU intensive. This leads to the conclusion that, given (I) and (II), a third condition must apply

- III A simulation involving a large number of particles (N_p) yet with only 2 physical particles colliding per MC cycle can become very inefficient. Therefore, (small) particles should be grouped together – i.e., considered as a single unit – during the collision.

Due to the duplication mechanism there is a natural way to group particles together. In equation (7) it was already seen that identical particles can be effectively absorbed in the calculation of the collision rates. We will now take this idea further and write the g_i 's (i.e., the occupation number of species i) as

$$g_i = w_i 2^{z_i}, \quad (8)$$

where w_i is the group number and z_i the *zoom number* (an integer equal to zero or larger) of species i . Thus, instead of tracking g_i particles, we now simulate only w_i groups, each containing 2^{z_i} particles. It is then the number of groups (N_g) that determines the collision rate per cycle. Because collisions are now between two groups of particles, involving 2^{z_i} and

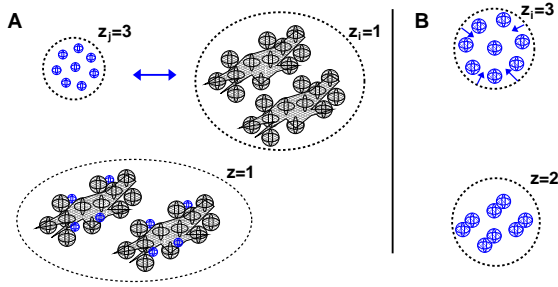


FIG. 2.— Illustration of collisions in the grouping algorithm. (A, *top*) Two groups with $z_j = 3$ and $z_i = 1$, respectively, collide. The zoom factors (z_i) give the number of particles within the group (2^{z_i}). Less massive particles have higher zoom factors than particles of higher mass. All particles within the group are identical. (*bottom*) Assuming sticking, the collision product is obtained by equal distribution of the collision partners. The end-product is a new species consisting of one group where the zoom factor corresponds to the lowest z of the progenitor particles. (B) Same, but for in-group collisions.

2^{z_j} physical particles, coagulation is speeded up significantly. That is,

$$N_g = \sum_i N_s w_i; \quad N_p = \sum_i w_i 2^{z_i}, \quad (9)$$

and we can, with a limited number of groups (N_g), represent large numbers of physical particles (N_p) by choosing appropriately high zoom factors z_i . This satisfies (I) and (II) and since each collision is now between two groups involving many particles we also avoid the inefficiency described in (III). Note that the base factor of 2 in equation (8) is arbitrary; however, it must be an integer (g_i is integer) and 2 is probably most convenient. More critical are the choices for the zoom factors, z_i : these must be chosen to reflect the nature of the system that is under consideration (see § 2.3).

A group collision is illustrated in Fig. 2a. The first group is magnified three times ($z_j = 3$), and the second has $z_i = 1$ ($z_j \geq z_i$ in this and all other examples). The groups thus consist, respectively, of 8 and 2 identical particles. During the collision, assuming sticking, the particles are equally distributed over the groups so that we end up with one group consisting of two identical particles. The mass of the new particle then becomes $m_i + 4m_j$ and the zoom factor $z = z_i = 1$ (the lowest of the two; the grouping may always be adjusted if needed). Note that in Fig. 2a only particles of different groups collide: out-of group collisions. However, if their λ_{ii} do not vanish, particles within a group can also collide: in-group collisions. This procedure is illustrated in Fig. 2b. The zoom number decreases by one in the final configuration (group collapse).

To generalize, for out-of-group collisions with $z_j \geq z_i$ every i -particle collides with $2^{z_j - z_i}$ j -particles. In total there are 2^{z_j} collisions, whereas for in-group collisions (Fig. 2b) there are $2^{z_i}/2$ collisions. Therefore, the collision rates of equation (7) – the rates corresponding to one collision – are divided by 2^{z_j} and $2^{z_i - 1}$, respectively, i.e.,

$$\lambda_{ij}^G = g_i g_j C_{ij} / 2^{z_j} = w_i w_j 2^{z_i} C_{ij} \quad (z_i \leq z_j, i \neq j);$$

$$\lambda_{ii}^G = \frac{1}{2} g_i (g_i - 1) C_{ii} / 2^{z_i - 1} = w_i (w_i 2^{z_i} - 1) C_{ii} \quad (i = j) \quad (10)$$

give the collision rates between two groups. Thus, although collisions between groups are less likely (the rate is decreased by a factor 2^{z_j}), this is ‘compensated’ by the 2^{z_j} particle-collisions that occur inside the group.

However, grouping does introduce various sources of systematic errors:

1. the outcome of a collision between two groups is fixed; in reality, a distribution of configurations will be present;
2. it is assumed that the grouped collision rates obey Poisson statistics like the single particle collision rate;
3. the individual collision rates $\{C_{ij}\}$ do not change *during* the group collision process to ensure that equation (10) is applicable.

A fixed configuration for the outcome of the grouping process is required because it prevents (unwanted) proliferation of the diversity of species, which would increase the N_s parameter and strain CPU requirements. However, the diversity of the simulation is already specified by the N_s parameter and we may simply check how N_s affects the accuracy of the collisional process. This point is similar to the second concern where all collisions are forced to occur at one point in time. Also, note that we must use the memoryless property of Poisson statistics; otherwise all collision rates must be adjusted at each timestep. However, it can be shown that, provided the collision rates stay constant, the grouped rates in equation (10) will conserve the mean of the particle distribution if $i \neq j$ (see Appendix A). The last assumption is the weakest: the individual rates between the particles in principle change after each individual collision as the collision affects the properties of the particles. Therefore, expressions as $g_i g_j C_{ij}$ should actually be averaged over the number of collisions that make up the group process. This is of course a very tedious procedure with which we would go back to the one-by-one treatment of the particles. Rather, we prefer to introduce a new parameter, f_ϵ , that limits the amount of mass accretion onto a particle during the group collisions, thereby strengthening the approximation that the C_{ij} stay constant during the group collision process.

Choosing the particle labels again such that j has the high zoom factor ($z_j \geq z_i$) the fractional mass accretion on a single i particle is $\Delta m_i / m_i = 2^{z_j - z_i} m_j / m_i$. From the preceding discussion it is this quantity that should be limited from below by a fixed value, f_ϵ . One way to implement this is to reduce the j -particles that take part in the collision. For example, in Fig. 2a this could mean that only 2 j -particles are involved, i.e., only a fraction of the particles in a group. Let this fraction be denoted by 2^{-N_ϵ} with $N_\epsilon \geq 0$ integer. Then, only $2^{z_j - z_i - N_\epsilon}$ j -particles are involved in the collision and the mass of an i particle increases by $2^{z_j - z_i - N_\epsilon} m_j$, i.e.,

$$\frac{\Delta m_i}{m_i} = \frac{2^{z_j - z_i - N_\epsilon} m_j}{m_i} \leq f_\epsilon. \quad (11)$$

Solving for the group splitting factor, N_ϵ , gives

$$N_\epsilon = \left\lceil -\log_2(f_\epsilon 2^{z_i} m_i / 2^{z_j} m_j) \right\rceil, \quad (12)$$

where the square brackets denote truncation to integer numbers ≥ 0 . Consequently, the collision rates, instead of equation (10), now become

$$\lambda_{ij}^G = g_i g_j C_{ij} / 2^{z_j - N_\epsilon} = w_i w_j 2^{z_i + N_\epsilon} C_{ij}; \quad z_i \leq z_j, \quad (13)$$

(the collision rates for in-group collisions do not change since $N_\epsilon = 0$). Another consequence is that the group numbers are real numbers, i.e., since a fraction 2^{-N_ϵ} of the j -group is removed, its group occupation number decreases accordingly, $w_j \rightarrow w_j - 2^{-N_\epsilon}$. The $\{g_i\}$ cannot become fractional, though, since this would physically split a particle. Also, a situation where $0 < w_i < 1$ is not allowed, because $\Delta w = 1$ for the particle of the lowest zoom factor; when it arises we ‘demagnify’ the group ($z_i \rightarrow z_i - 1$) such that $w_i \geq 1$ (see below, § 2.4).

2.3. Choosing the zoom factors

We have implemented the grouping method in such a way that the zoom factors, are free parameters, and there is indeed considerable freedom to choose (an algorithm for) the $\{z_i\}$. However, it is also an important choice as it will determine how *efficient* the Monte Carlo simulation becomes. Here, with “efficient” it is meant that the number of species N_s should not (suddenly) increase. A way that observes this constraint fairly well is to divide the mass inside the simulations equally over the groups. Additionally, we present a more general method, in which the zoom factors are determined by the particle distribution.

2.3.1. Equal mass approach

In this method, we assume there is a characteristic mass m^* after which the zoom factors are determined. Particles of mass larger than m^* are ungrouped ($z = 0$), whereas particles of mass $m < m^*$ have zoom factors such that the group mass is around m^* , i.e.,

$$z_i = \left\lceil \log_2(m^*/m_i) \right\rceil. \quad (14)$$

In the case of constant- V simulations and monodisperse initial conditions of unity mass the simulation then starts with N/m^* groups, where N is the initial number of particles. Therefore, N/m^* is a measure for the number of species parameter N_s . The target value of N_s , N_s^* , and N then determine the value of m^* . For the constant- N_s simulations, where the mass in the simulation varies, the mass peak m_p is a natural value for the characteristic mass, $m^* = m_p$. It forces most of the groups to represent particles around the mass peak, with those of higher mass ungrouped.

Thus, the equal mass approach resolves the mass density of the distribution, instead of the particle density as in conventional MC-methods. Once species becomes insignificant in terms of mass, $m_i 2^{z_i} < m^*$, occupancy number approaches unity ($w_i = 1$) and the species will be “removed” from the simulation when it collides at the next collision, completely independent of the actual number of particles a group contains.

2.3.2. Distribution approach

A different way to choose the zoom factors is to force the distribution to be resolved over its entire mass range. This can, for example, be done by dividing the distribution into exponentially-spaced mass bins and to choose the zoom factors such that there are a fixed number of groups per bin. Here the bins are exponentially distributed between a lower mass (most often the initial mass, m_0) and the largest mass M_{L1} present in the simulation. For example, let N_k be the total occupancy number of bin k , then the zoom numbers are determined by the requirement that an equal number of groups is present in each bin, i.e., $N_k/2^{z_k}$ is constant for each bin k . There is no constraint on the number of species per bin, but as this quantity is always less than the number of groups it is limited as well. For a smoother progression of zoom factor with mass we allow for a linear interpolation within each bin, such that the zoom factors become a continuous, piecewise, and decreasing function of mass, $z(m)$. Thus, z is now determined, not by the particle or mass density, but by the sheer occurrence of particles of a certain mass. The bins and zoom factors are dynamically adjusted, as the simulation progresses, and the distribution changes.

The distribution method has the consequence that (high- m) fluctuations are resolved very well, even if they are completely insignificant by number and mass. The zoom factors of these particles are much reduced in comparison with the equal mass method. For example, if the $m_0 = 1$ particles are still at zoom levels of z_0 situations where $2^{z_0}m_0 \gg 2^{z_i}m_i$ are frequently encountered. A collision between a massive and a $m = 1$ group then takes place at a high value of N_e (eq. [12]) and collision rates involving high- m groups increase (eq. [13]). Since each collision creates a new species, the number of species (N_s) also increase rapidly – too rapid, in fact – and we are forced to merge species (§ 2.1.3). Merging of species is therefore a necessary feature of the distribution method.

2.4. (De)magnification

In the grouping method the zoom factors of the groups are determined in accordance with one of the two methods above. The zoom factors will therefore change during the simulation. When z_i increases ($z_i \rightarrow z_i + 1$) we speak of *magnification*, whereas when z_i decreases the group is *demagnified* ($z_i \rightarrow z_i - 1$). This dynamical adjustment of the zoom factors is the main reason why the method is capable of achieving very high dynamic range. Demagnification furthermore occurs when the occupancy number of a group becomes fractional, $0 < w_i < 1$. However, we note again that the actual number of particles of a species, g_i , is always integer.

There is a small price to pay in terms of overhead for these re-adjustments. Because the $\{z_i\}$ are an argument of the collision rates (eq. [13]) these rates (or, more accurately, the $\{\lambda_i\}$ quantities, as these are the ones that are stored in the computer’s memory, $\sim N_s$ in total) have to be adjusted at each (de)magnification event. This may slow down the simulation by a factor of 2. Finally, we note that (de)magnification is, within the context of the grouping algorithm, an exact procedure; collision rates are not approximated and the total mass of the simulation stays conserved.

2.5. Grouping Summary

We have outlined an approach for a Monte Carlo coagulation scheme that can involve many particles – allowing the growth to be orders of magnitude – yet avoids the severe computational problems that would naturally arise with a single particle per collision treatment. In Fig. 3 we summarize the main steps of the algorithm, within the context of the grouping method. At its core lies the approximation that many identical particles are present, due to monodisperse initial conditions or the duplication mechanism, which can be efficiently grouped together or (in the distribution method) merged. Collisions are now between two groups of, possibly, many particles and this significantly speeds up the computation. While it is clear that such grouping inevitably approximates the collision process, we have advocated a fundamental shift in the implementation of MC methods, i.e., to focus on the particles in a distribution that really matter – e.g., those around $m = m_p$ – and we anticipate that the benefits resulting from the new emphasis will more than justify the approximations inherent to the grouping method. In any case, to validate the new method a comparison must be made with the analytic solutions of the classical kernels, after which the algorithm can be applied to more physically interesting kernels.

3. RESULTS

1. Setup state vector

species	group number	zoom factor	Particle properties	
			mass	(other properties)
1	w_1	z_1	m_1	...
2	w_2	z_2	m_2	...
...
N_s	w_{N_s}	z_{N_s}	m_{N_s}	...

Note: w_i can be real but not less than unity ($w_i \geq 1$); z_i is integer. Number of particles of species i , $g_i = 2^{z_i} w_i$, is also integer.

2. Calculate group collision rates

$$\lambda_{ij}^G = w_i w_j C_{ij} / 2^{z_i + z_j} \quad (\text{if } i \neq j);$$

$$\lambda_{ii}^G = \frac{1}{2} w_i (w_i - 1) C_{ii} \quad (i = j, \text{ In-group collision})$$

Store partial rates, $\lambda_i = \sum_{k=i}^{N_s} \lambda_{ik}^G$

Store total rate, $\lambda_{\text{tot}} = \sum_i \lambda_i$

Where:

- C_{ij} : individual collision rate between two particles, K_{ij}/V . The collision kernel $K_{ij} = \sigma_{ij}(\Delta v)_{ij}$ depends on mass and other particle properties.
- V : simulation volume
- N_ϵ : group splitting factor that regulates mass accretion of j -particles on i -particles (here: $z_j \geq z_i$) to avoid excessive instantaneous mass load. It is defined as

$$N_\epsilon = \left\lceil -\log_2(f_\epsilon 2^{z_i} m_i / 2^{z_j} m_j) \right\rceil, \quad (1)$$

where f_ϵ is the maximum allowed fractional increase in mass of the larger particle i .

- N_s : total number of species

3. Pick collision particles

This follows the “full conditioning” method of Gillespie (1975).

time increment: $\Delta t = \lambda_{\text{tot}}^{-1} \ln(1/\bar{r})$;

i – particle: $\sum_{k=1}^{i-1} \lambda_k \leq \bar{r} \lambda_{\text{tot}} \leq \sum_{k=1}^i \lambda_k$;

j – particle: $\sum_{k=1}^{j-1} \lambda_{ik}^G \leq \bar{r} \lambda_i^G \leq \sum_{k=1}^j \lambda_{ik}^G$.

where \bar{r} are (different) random deviates.

4. Perform collision

- (Choose/rearrange i, j such that $z_j \geq z_i$).
- Determine the group-splitting factor N_ϵ and the collision multiplicity
- Each i -particle collides with $N = 2^{z_j - z_i - N_\epsilon}$ j -particles.

Perform the group collision: the sequential collision of N j -particles with the i -particle. Determine the new (structural) properties of the particles created.

5. Update collision rates, state vector

- Reduce group-number of collision species: $w_i \rightarrow w_i - 1$, $w_j \rightarrow w_j - N_\epsilon$.
- If w_i or w_j becomes 0: remove this row from the state vector ($N_s \rightarrow N_s - 1$) and remove entry from $\{\lambda_k^G\}$.
- Add the new particle to the state vector at index $N_s + 1$ ($N_s \rightarrow N_s + 1$).
- Update the $\{\lambda_k^G\}$. For every entry k look whether the rates $\lambda_{ik}, \lambda_{jk}$ should be subtracted (if $k \leq i$, resp., $k \leq j$) and add rate λ_{qk} if $k \geq q$. Recalculate λ_{tot} .

6. Adjust $\{z_i\}$, merge species, choose duplicates

These steps are necessary for a smooth progression of the program, but do not have to be processed at every cycle. Adjust collision rates, $\{\lambda_i\}$ accordingly. Zoom factors can be adjusted up or down:

- magnification: zoom factor of species is increased, $z_i \rightarrow z_i + 1$ and $w_i \rightarrow w_i/2$.
- demagnification: zoom factor of species is decreases, $z_i \rightarrow z_i - 1$ and $w_i \rightarrow 2w_i$.

Note: We have outlined two methods to regulate $\{z_i\}$. (I) Equal mass method: z_i is related to the mass peak and requires duplication; (II) distribution method: z_i determined by the distribution. In both methods the number of species parameter N_s must be constrained.

- Merging: properties of similar species are averaged to one species. This limits the number of species in a mass interval.
- Duplication (constant- N_s simulations only): pick species- k randomly from statevector (with weights $\{w_i\}$) and add to it ($w_k \rightarrow w_k + 1$). Update collision rates $\{\lambda_i\}$.

Note: Duplication does not directly increase N_s but increasing group numbers make occurrence of empty rows less likely. If particles are duplicated the simulated volume V also increases. Constant- V simulations do not require duplication.

End of cycle → return to 3

Fig. 3.— Chart outline the steps required to implement the grouping algorithm.

In this section the grouping method – both the equal mass approach, as well as the distribution method – is tested in situations that require a high dynamic range. Its performance is then discussed. First, the grouping method is tested against the sum kernel. We show that both the equal-mass and the distribution methods are capable of following the analytic distribution, whereas conventional MC methods (without grouping) will fail. The grouping method is also tested against the product kernel, which is one of a class of kernels that produces runaway-growth behavior. Here, we will see that it is important to accurately follow the fluctuations of the mass distribution, which can only be done by the distribution method. More runaway-models are discussed in § 3.3. Finally, we also

perform simulations including a simplified form of fragmentation, where the particles are catastrophically fragmented at the largest scale m_{frag} to be injected back at the smallest scale m_0 (§ 3.4). We test whether the grouping method can also be applied in these cases. An overview of the simulations is given in Table 3, including the adopted numerical parameters. These are chosen such that the CPU-time per simulation is typically several hours on a modern desktop machine (1.8 GHz clock speed). Similar CPU-times apply for the choice of the numerical parameters of the simulations including grouping.

For the product and other runaway kernels, the system will after a certain time be dominated by a single particle. Hence,

TABLE 3
LIST OF SIMULATIONS

Kernel/description (1)	Figure ref. (2)	Type (3)	Grouping (4)	N_s^*/N_g^* (5)	N (6)	f_ϵ (7)	N_{sim} (8)
sum kernel	Figure 4	constant- N_p	none	10^5	...	10^{-4}	40
sum kernel	Figure 5a	constant- N_s	equal mass	2×10^4	...	10^{-4}	40
sum kernel	Figure 5b	constant- \mathcal{V}	distribution	10^4	10^{100}	10^{-4}	40
product kernel	Figure 6	constant- \mathcal{V}	equal mass	2×10^4	10^{20}	10^{-4}	40
product kernel	Figure 7	constant- \mathcal{V}	distribution	10^3	10^{20}	10^{-4}	40
runaway kernels	Figure 8	constant- \mathcal{V}	distribution	$\sim 10^4 - 10^5$	$300 - 10^{160}$	10^{-4}	10-100
sum: fragmentation	Figure 9	constant- N_s	equal mass	2×10^4	...	10^{-3}	1 ^a
constant: fragmentation	Figure 10	constant- N_s	equal mass	10^4	...	10^{-3}	1 ^a

NOTE. — Summary of all simulations. (1) Collision kernel. (2) Figure reference. (3) Simulation type: fixed or adjustable volume. (4) Grouping method: outlines in which way the zoom factors, $\{z_i\}$, are chosen (no grouping, equal mass, or distribution method). (5) Target number of species (equal mass method), or groups (distribution method) in simulation. When multiple values apply the largest is given. (6) Initial number of monomers, or total mass in simulation for constant- \mathcal{V} simulations. (7) Fractional mass accretion. (8) Number of simulations performed. The initial distribution is monodisperse in all simulation with $m_0 = 1$.

^a In fragmentation simulations a steady state is reached in which the distribution is averaged ~ 100 times during the same simulation.

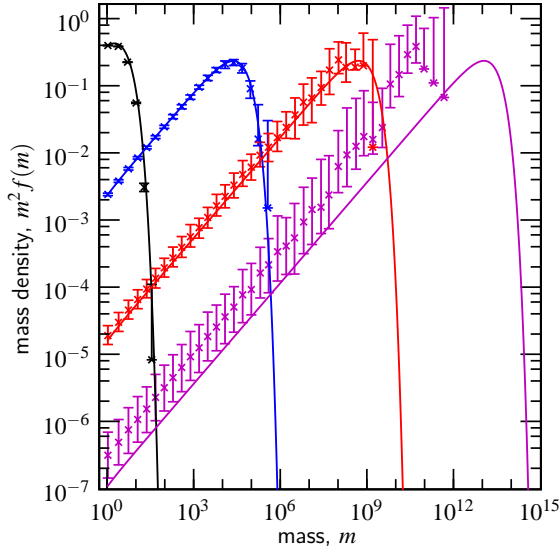


FIG. 4.— Results for the sum kernel, $K_{ij} = \frac{1}{2}(m_i + m_j)$, without grouping for $N_p = 10^5$. Distributions are plotted at dimensionless times $t = 1, 10, 20$ and 30 and error bars show the spread over 40 simulations. Particles are binned at each factor of two in mass. The correspondence with the analytic results (solid lines) deteriorates over time. After $t \approx 20$ the mass peak is no longer resolved by the simulation.

no self-similar solutions exist and in order to study the evolution properly, the simulation volume of these systems must be well defined. These simulations are therefore modeled using constant-volume simulations. In most cases the distribution method is used, which also requires a fixed simulation volume. In the other cases, we may assume that a small volume is representative of the parent distributions, which can then be modeled using constant-number simulations. The only exception is Fig. 5b, where we have used the distribution method in order to compare it to the results of the equal-mass method simulations of Fig. 5a.

3.1. Test case I: the sum kernel

Before presenting the results of the grouping mechanism, a case without the grouping method is considered first, illustrating some of the ‘deficiencies’ mentioned in § 1. We will not consider the constant kernel, because in the constant kernel the dynamic range ($m_p/\langle m \rangle$) stays confined and a grouping method is not required. In Fig. 4 the number of simulation

particles, N_p , is kept constant at 10^5 . The distribution function is plotted at various times during its evolution, $t = 1, 10, 20$ and 30 . Times are dimensionless, i.e., in units of $(K_{00}n_0)^{-1}$ where K_{00} and n_0 are the kernel and number density corresponding to the initially monodisperse conditions of particles of mass $m_0 = 1$. In Fig. 4, as well as in all other figures that show distributions from the MC-simulation, $f(m)$ is determined using exponentially-spaced bins every factor of 2 in mass. Each point shows the average and spread of in total 40 model runs. The analytic distribution functions at these times are given by the solid curves.

Initially, $t < 10$, the agreement between the MC simulation and the analytic curve is very good. The mass range in the simulation is still rather modest and the 10^5 particles suffice to represent all relevant masses. However, while the peak mass progresses steadily to higher m , the majority of particles stays behind at low masses; that is, although both $\langle m \rangle$ and m_p evolve exponentially with time (Fig. 1), $m_p/\langle m \rangle$, which is a measure of the dynamic range, also increases exponentially. Since MC-methods sample the number density, increasingly fewer particles become associated with the mass peak, and the simulation loses numerical resolution. The effects of these low-number statistics are a bit alleviated by averaging over many different simulation runs but it is not merely a statistical deterioration: the mass peak dominates the collisional sweep-up of the system and the fluctuations caused by an inaccurate characterization blow up. There is especially a trend to lag the analytic solutions because, it is impossible to ‘recover’ from a lagging distribution, because the duplication mechanism does not duplicate the particles that are not present. At $t = 20$ these problems already become apparent when the spread is very high, especially around the mass peak, and there is a slight tendency to lag the analytic curve. At $t = 30$ the effects are disastrous: 10^5 particles are simply too few to sample a large enough volume to represent the large particles.

Due to the duplication mechanism many duplicates are created and N_s slowly decreases. This somewhat speeds up the computation at larger times since collisional rates scale with N_s^2 . A more accurate result will be obtained by keeping N_s constant (Ormel et al. 2007). However, now N_p increases and the simulation suffers computational freeze-down as it progresses. Fundamentally, either approach – the constant N_p or constant N_s – is the same since each cycle only treats one collision and each collision is equally important.

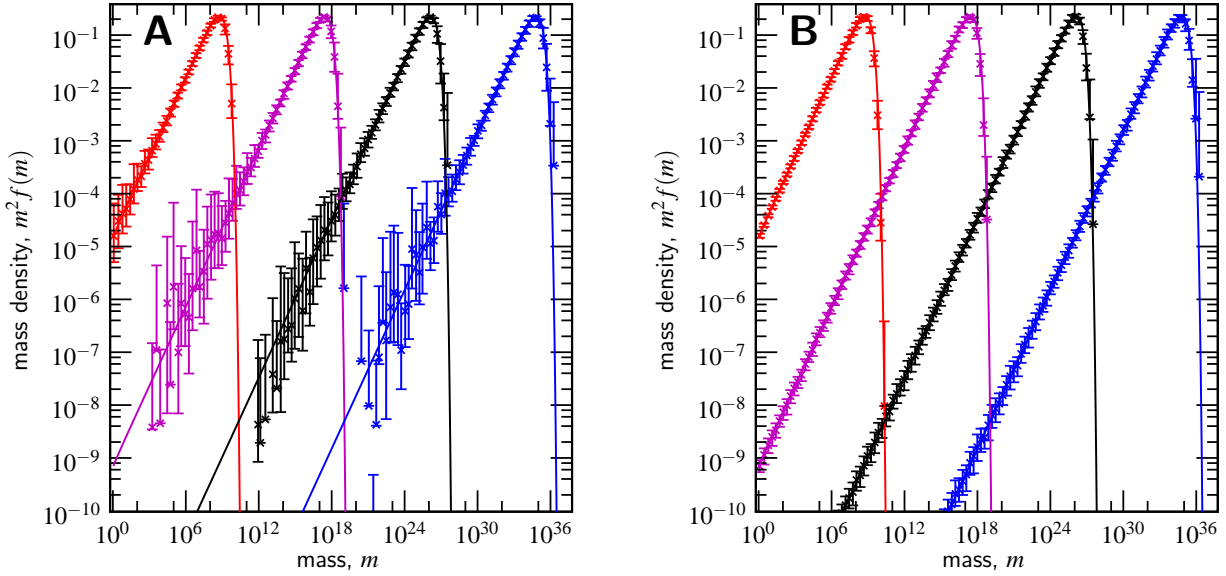


FIG. 5.— Results for the sum kernel using the grouping algorithm. Particle distributions (symbols) are shown at times $t = 20, 40, 60$ and 80 . Particles are binned at each factor of two in mass and results from 40 separate simulations are combined. The spread in the simulations is denoted by the error bars. (A) Equal mass method, $N_s = 20000$. Thanks to the grouping algorithm most of the groups are centered on the mass peak, which favours an accurate computation of the collisional evolution, without introducing systematic errors. (B) distribution method with $N_g = 10000$, distributed over 20 bins. The zoom factors are adjusted to resolve the distribution over its entire range.

Clearly, this ‘socialist’ approach is ineffective as Fig. 4 illustrates: the more massive particles deserve more attention than their smaller brethren.

Results with the grouping method applied are shown in Fig. 5. In Fig. 5a we used the equal mass method for the groups (each group has a mass of $\sim m^* = m_p$ if $m < m_p$ and particles with $m > m_p$ are not grouped) and choose the zoom factors accordingly (see eq. [14]). The number of species, N_s , is stabilized at a target number of $N_s^* = 2 \times 10^4$ through the duplication mechanism. That is, if N_s due to coagulation falls below N_s^* , duplicates are added to the system until $N_s \geq N_s^*$. The simulation total mass and volume then increase, but their ratio (the mass density) is fixed. As m_p becomes larger during the simulation, zoom factors of particles that do not take part in collisions will increase, according to equation (14). This ‘magnification’ causes the occupancy numbers w_i to stay low, despite the ongoing duplication events. When w_i reaches 1, the species will disappear at the next collision, freeing up this slot for more massive particles. The grouping thus forces the particles to act collectively. This procedure prevents the systematic errors (N_p constant, Fig. 4) or computational freeze down (constant N_s) of the ‘1 collision per cycle’ approach of the conventional MC-methods.

Figure 5a shows that the grouping mechanism resolves the mass peak well during its entire simulation. The agreement between the analytic and numerical method remains very good during the entire simulation. Figure 5a proves that once the mass peak is resolved, the simulation will accurately compute the collisional evolution, and systematic errors as in Fig. 4 are prevented. Thus, the resolution of the mass peak is key. However, at lower mass densities the groups disappear from the simulation as they do not have sufficient mass.

These low- m statistics can be accurately followed when we switch the method for assigning the group’s zoom factor $\{z_i\}$ to the distribution method, in which the zoom factors are adjusted such that there are an equal number of groups per exponential mass interval. The zoom number for the low- m particles is then much lower in comparison with the equal mass

method. Figure 5b shows that the mass density as well as number density are accurately followed over the entire distribution. It can be seen in Fig. 5b that the match to the high- m tail slightly deteriorates over time. This is a consequence of the equal spacing in log space of the bins that determine the zoom factors. The high- m tail, which falls off exponentially, becomes associated with only a single bin, and is then less well resolved. A more flexible placement of the bins is expected to give an even better correspondence. Note again that the bins are only needed to determine the zoom factors; there’s no actual binning of the particles in the MC-simulation.

Although Fig. 5b more precisely follows the distribution – i.e., it both follows the mass density as well as the number density – it requires to *merge* species in order to restrict their numbers (see § 2.1.3). For these test cases, in which only mass is a parameter, Fig. 5b shows that this is not problematic. However, when structural parameters are involved, the merging procedure is less obvious; i.e., are species of comparable mass merged or, say, species of comparable porosity? The merging principle is, therefore, somewhat against the spirit of MC-simulations, as there is no physical equivalent of ‘merging’ in nature. Thus, for a MC simulation with many structural parameters, where it is not required to resolve the number density, Fig. 5a is probably the preferred method. In the other test cases of this section we will therefore first pursue whether these can be modeled by the equal-mass method.

3.2. Test case II: the product kernel

The kernel $K_{ij} = m_i m_j$ is the only runaway kernel with an analytic solution. In models that show runaway-behavior a particle will appear that separates from the continuous distribution to consume, in a relatively short time, all the available mass. This separation is an intrinsic feature of the kernel and is not caused by a poor sampling of the distribution. The point at which this separation occurs, t_R , depends on the properties of the kernel and, in some cases, on the initial number of particles (N). In the case of the product kernel it always occurs at $t = 1$, independent of N . The runaway growth properties of other kernels are discussed in § 3.3.

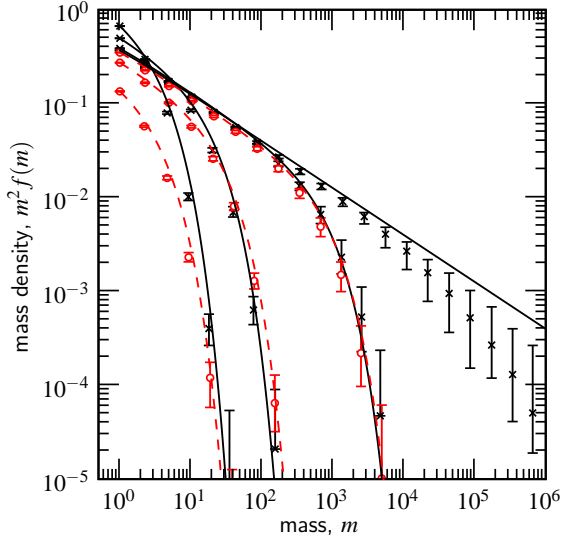


FIG. 6.— Results for the product kernel, $K_{ij} = m_i m_j$, with $N = 10^{20}$. The simulated distribution is plotted at times $t = 0.4, 0.7, 0.95, 1.0$ (from left to right, crosses) and $t = 1.05, 1.3$ and 2.0 (from right to left, open circles). The analytic solution is plotted by solid curves for $t \leq 1$ and by the dashed curves for $t > 1$ (the $t = 0.95$ and $t = 1.05$ curves almost overlap). Error bars denote the spread in the 40 simulations. After $t \approx 1$ a runaway particle of mass $m \sim N$ appears that consumes most of the mass in the system.

In § 1 it was noted that the Smoluchowski coagulation equation becomes invalid in the case of the runaway kernel: it keeps assuming that the distribution is continuous and it is therefore ill-suited to describe a discrete system caused by the runaway particle. Wetherill (1990) has adjusted the original (incorrect) solution of Trubnikov (1971) by inclusion of a term for the runaway body in equation (1). Although not mathematically rigorous, this approach captures the physics of the runaway: as soon as the (incorrect) numerical solution exhibits a $f(m) \propto m^{-5/2}$ power-law tail, the interaction between the higher mass bins causes the runaway growth. In the Monte Carlo simulations we also expect that the appearance of a power-law tail is a signature that runaway growth is incipient.

In Fig. 6 the distributions corresponding to the product kernel with $N = 10^{20}$ are plotted at dimensionless times $t = 0.4, 0.7, 0.95, 1.0, 1.05, 1.3$ and 2.0 , using the equal mass method. Since we start the simulation already with a huge number of particles, large zoom numbers are required. We have fixed the mass peak at $m_* = N/10^4$, which causes the particles to be sufficiently magnified, except for the runaway particle. (The 10^4 factor may be thought of as a fudge parameter akin to N_s or N_p in the constant- N simulations.) Theoretical curves correspond to the Trubnikov (1971) solutions which, as Wetherill (1990) showed, are still valid for the continuous part of the distribution. At $t = 1$, runaway growth is expected and the continuous power-law distribution reaches its furthest extent: the $-5/2$ power-law exponent. After the runaway particle has separated the continuous distribution shrinks, especially in the $-x$ direction since the more massive particles are consumed rapidly by the runaway body, which itself is not shown in Fig. 6.

The runaway growth is generally well modeled by the Monte Carlo method. The distribution becomes most sensitive near $t = 1$ where a small change in time means a big change in the high- m distribution function. Due to the stochastic behavior of the Monte Carlo method, the precise timing of the runaway event does not exactly coincide with

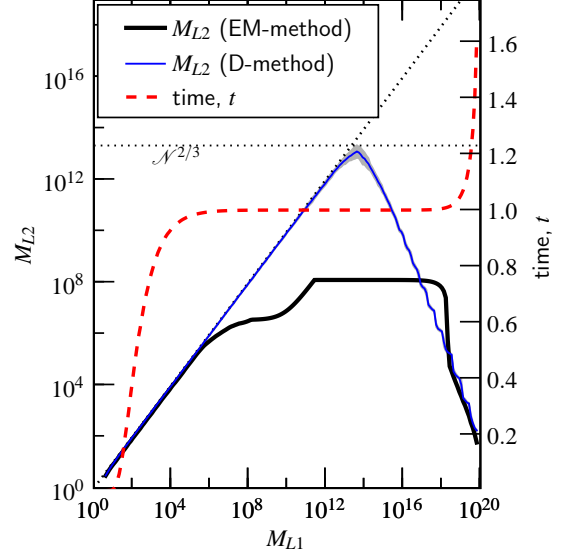


FIG. 7.— Statistics for the simulations involving the product kernel. As function of the mass of the largest species, M_{L1} , are shown: (i) the mass of the second-largest species, M_{L2} , according to the equal-mass method (thick solid curve) and the distribution method (thin solid curve); and, (ii) the time (dashed curve). Values are averaged over 40 runs. The auxiliary lines $M_{L2} = M_{L1}$ and $M_{L2} = N^{2/3}$ are also given (dotted lines).

$t = 1$. Therefore, the $t = 1$ distribution lags the theoretical curve since at $t = 1$ either runaway growth did not yet occur (and the distribution still has to reach the power-law curve) or has already happened (and the distribution is on its way back from the power-law curve). In Fig. 7 the relation between the two most massive groups is denoted by the thick solid line, together with the simulation time t as function of M_{L1} . Figure 7 shows that until $t = 1$ there is little difference between the mass of the most massive particle – actually group – M_{L1} and the second-most massive group M_{L2} . However, as runaway is reached at $M_{L1} \sim 10^5$ the masses diverge very quickly.

These results, however, are in disagreement with the theoretical study of Tanaka & Nakazawa (1994). Comparing the solutions to the stochastic and the statistical equation they find that, in the case of the product kernel, the statistical equation (eq. [1]) becomes invalid after $M_{L1} \sim N^{2/3}$. It is only at this point that M_{L1} and M_{L2} should start to diverge. However, in the equal-mass method (Fig. 7, thick line) the divergence happens much earlier: M_{L2} never reaches beyond $10^8 \ll (10^{20})^{2/3}$. This discrepancy can be explained as an artifact of the equal mass method. Because each group is still of considerable mass, $m^* \sim 10^{16}$, runaway growth is just the collapse of one group by a continuous series of in-group collisions (Fig. 2b); that is, until $z_i = 0$, the M_{L1} group does not correspond to a single particle and therefore never really separates. It is thus not a proper modeling of a runaway process. In order to accurately follow the behavior of the particles during runaway growth, the groups should be demagnified much more rapidly than equation (14) allows. It is perhaps surprising that the equal-mass method still works fine – i.e., its distribution agrees with the analytical case (Fig. 6) – despite the fact that it does not resolve the runaway particles. However, in general this is not the case: the fluctuations present at high- m require more resolution than what is attributed to them through the equal representation of mass density (eq. [14]).

In Fig. 7 the M_{L1}, M_{L2} statistics of the product kernel are recalculated with the distribution method for the zoom factors (thin solid line). According to the Tanaka & Nakazawa (1994) prediction M_{L1} and M_{L2} should then separate at $M_{L1} \sim N^{2/3}$.

TABLE 4
RUNAWAY MODELS

type	conditions	growth mode
(i)	$\nu \leq 1, \beta \leq 1$	orderly growth
(ii)	$\nu \leq 1, \beta > 1$	runaway growth at t_0
(iii)	$\nu > 1$	instantaneous runaway growth ^a

NOTE. — Properties of runaway models according to the study of Lee (2000). The exponents ν and μ give the mass dependence of the kernel on a heavy and light particle, respectively, with $\beta = \nu + \mu$ the total mass dependence, $K_{ij} \propto m_i^\nu m_j^\mu$ ($m_i \gg m_j$).

^a Instantaneous in the limit $N \rightarrow \infty$.

This is exactly the trend seen in Fig. 7. We therefore conclude that the distribution method is the appropriate choice to compute the collisional evolution of runaway kernels.

3.3. Strong Runaway kernels

There is an extensive literature on the phenomenon of runaway growth, especially mathematical ($N \rightarrow \infty$), where the phenomenon is termed *gelation*. In general, three growth modes can be considered (see Table 4): (i) kernels where runaway growth does not appear, as in the constant or sum kernel; (ii) kernels that exhibit growth after some characteristic time, as in the product kernel; and (iii) kernels in which gelation is *instantaneous*¹. The occurrence of instantaneous gelation for the class of kernels $K \propto (m_i m_j)^\nu$ with $\nu > 1$ was conjectured by Spouge (1985) on the basis of MC-coagulation experiments and was proved by Jeon (1998). Lee (2000) illustrates these distinct modes of solution by a simple coagulation model, in which the distribution is approximated by a characteristic mass m , e.g., the peak mass, and a test particle of mass $M > m$. Particles of mass m dominate the total mass and hence $n(m) \propto m^{-1}$. Furthermore, the coagulation kernel is characterized by two exponents, ν and μ , that give the dependence of $K(m_i, m_j)$ in the case of a heavy (i) and a light particle (j), i.e., $K_{ij} \propto m_i^\nu m_j^\mu$, for $m_i \gg m_j$. The total mass dependence is then given by $\beta = \mu + \nu$. For example, in the case of the product kernel $\nu = \mu = 1$ and $\beta = 2$. Setting the physical density equal to unity, the equations for the simplified merger problem become (Lee 2001)

$$\frac{dm}{dt} = \frac{1}{2} m^\beta; \quad (15)$$

$$\frac{dM}{dt} = m^\mu M^\nu. \quad (16)$$

These can also be combined to obtain

$$\frac{dM}{dm} = 2 \left(\frac{M}{m} \right)^\nu. \quad (17)$$

Equation (15) can be solved as

$$m = \begin{cases} \left[1 + \frac{1}{2}(1-\beta)t \right]^{1/(1-\beta)} & \text{for } \beta \neq 1; \\ \exp[\frac{1}{2}t] & \text{for } \beta = 1, \end{cases} \quad (18)$$

where we have for simplicity taken $m_0 = m(t=0) = 1$. Equation (17) can be solved in terms of m as

$$M(m) = \begin{cases} \left(M_0^{1-\nu} + 2m^{1-\nu} - 2 \right)^{1/(1-\nu)} & \text{for } \nu \neq 1; \\ M_0 m^2 & \text{for } \nu = 1, \end{cases} \quad (19)$$

¹ We note that in a physical situation where runaway growth tends towards instantaneity, causality should not be violated. As we will see, N , even in astrophysical circumstances, does not become so large that the process is instantaneous.

where $M_0 > 1$ is the initial mass of the test particle. Three qualitatively distinct growth modes follow from these equations: (i) if $\nu \leq 1$ and $\beta \leq 1$ both equation (18) and equation (19) have finite solutions at all t : orderly growth. (ii) If $\nu \leq 1$ and $\beta > 1$, equation (18) reaches a singularity at a time $t = t_R = 2/(\beta - 1) > 0$,² while equation (19) does not: both m and M become infinite at the same time t_R that does not depend on M_0 . (iii) If $\nu > 1$, equation (19) reaches infinity for a finite value of m . Furthermore, the gelation time *decreases* with increasing M_0 , $t_R \approx M_0^{1-\nu}/(\nu - 1)$. Then, if the initial mass in the system goes to infinity, a sufficiently strong fluctuation (large M_0) will always be present to satisfy the singularity condition for $M(t)$ even as $t \rightarrow 0$; this is the physical reason behind the instantaneous gelation conjecture.

Here, we will concentrate on the case $\nu > 1$ and test the instantaneous gelation conjecture for kernels $K = (m_i m_j)^\nu$ with $\nu = 2, 3$. Malyshkin & Goodman (2001) extend the simplified merger model discussed above to a model with only two kinds of particles – one predator and many preys – and one collisional process (the predator feeds of preys). This is the monotrophic model, e.g., one particle (‘mono’) that nourishes (‘trophin’) of the others. Considering this model Malyshkin & Goodman (2001) compute the probability that the predator occupies any finite mass, which decreases with increasing time. Consequently, the gelation time t_R is computed as

$$t_R \propto (\log N)^{1-\nu}, \quad (20)$$

where N is the initial number of particles present. Thus, the timescale for runaway growth decreases as function of the initial number of particles in the system, but only logarithmically. Equation (20) may be anticipated from the discussion above on the simple merger model. Consider a distribution of particle masses $N(m)$ at a fixed time. It is quite likely to expect that the tail of the distribution is exponentially suppressed as in the case with Poisson statistics, i.e., $N(m) \propto N \exp[-m]$ with N the total number of particles. Setting $N = 1$ we thus obtain $M_0 \sim n \sim \log N$. Hence, the logarithmic dependence on the number of particles. Malyshkin & Goodman (2001) test this model by Monte Carlo simulations and found qualitative agreement: the gelation time decreases as function of $\log N$, although the power-law exponent is not precisely that of equation (20): $1 - \nu$. However, the simulations of Malyshkin & Goodman (2001) were limited in their dynamic range; we will extend their study to much larger values of N to see whether equation (20) holds over a sizeable range of $\log N$.

Note that as conjectured earlier, a good census of the fluctuations is the key to an accurate modeling of the collisional evolution. Therefore, the grouping must then be implemented by the distribution method. There is, however, another catch. At $t = 0$ the number of groups of the monodisperse species is $w_0 \sim N_g^* \ll N$ and the collision rate for a collision between two monomers according to equation (10) is $\lambda_{11} \sim N w_0 C_{11} \sim w_0$. Then, the timescale for collisions involving monomers is $t_{11} = \lambda_{11}^{-1} \sim w_0^{-1}$, which can, however, become comparable to, or larger than, the runaway timescale of equation (20). Therefore, the monodisperse groups have to be demagnified such that $t_{11} \ll t_R$. By forcing the appropriate collision timescale for the monodisperse collisions, the fluctuations are adequately resolved, but it also means that the

² Note that this expression disagrees with the product kernel in which $\beta = 2$: the simplified merger model does not give the correct prefactor, see Lee (2001).

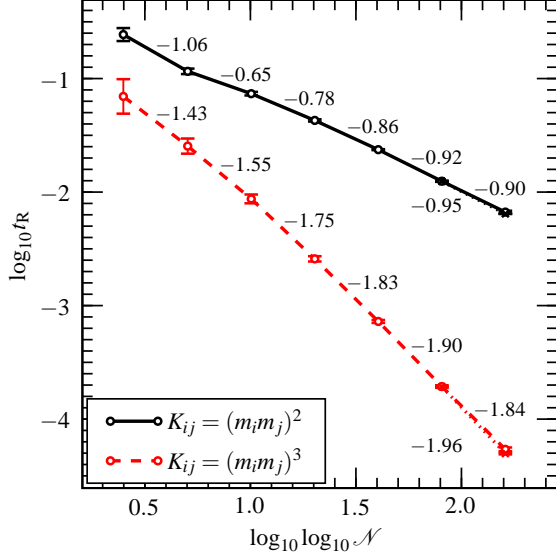


FIG. 8.— The runaway time t_R as function of the initial number of particles N for the kernel $K_{ij} = (m_i m_j)^\nu$ with $\nu = 2$ (solid line) and $\nu = 3$ (dashed line). Simulations are run at $N = 300, 10^5, 10^{10}, 10^{20}, 10^{80}$ and 10^{160} . The power-law index is indicated at every interval of $\log N$. The agreement with the theoretically expected value for the power-law index, $1 - \nu$, becomes better at higher N . At $N = 10^{160}$ a simulation at a higher numerical resolution gives an even steeper slope (indicated below the line) confirming the converging trend towards, respectively, -1 and -2 .

CPU-time increases, especially at large N . For the $\nu = 2$ simulations we used a number of $N_g^* = 60\,000$, whereas in the $\nu = 3$ simulations we take $N_g^* = 300\,000$. At $\nu = 3$, runaway occurs at a lower value of M_{L1} and we are allowed to start with a higher number of groups. However, the discrepancy between the simulations at different values of N becomes large in terms of CPU-time: the highest- N simulations take about half a day to finish.

The results for the kernels $K = (m_i m_j)^2$ and $K = (m_i m_j)^3$ are presented in Fig. 8. We extend the model of Malyshkin & Goodman (2001) to an unprecedented 10^{160} number of particles (about the square of the number of particles in the universe). To speed up the simulations at high N we have decreased the mass-ratio parameter f_ϵ to 10^{-3} (which we checked to be still sufficiently accurate). The number of simulations varies between $N_{\text{sim}} = 100$ at low N and $N_{\text{sim}} = 10$ at $N = 10^{160}$. We stop the simulations when the ratio between the two largest masses becomes $M_{L1}/M_{L2} \geq 10^4$, which is a measure of t_R . By this time the simulation has either already finished or has reached its asymptotic limit.

According to the analytic prediction (eq. [20], Malyshkin & Goodman 2001) the t_R - $\log N$ power-law is -1 and -2 for $\nu = 2$ and $\nu = 3$, respectively. In Fig. 8 the piecewise exponent is indicated (number above the lines), which shows a declining trend, except for the $N = 10^{160}$ point. To check convergence, we have additionally performed five simulations at a higher numerical resolution for $N = 10^{80}$ and $N = 10^{160}$ ($N_g^* = 2 \times 10^5$ for the $\nu = 2$ simulations and $N_g^* = 2 \times 10^6$ for the $\nu = 3$ simulations). For $N = 10^{80}$ it was found that the results had converged, but at $N = 10^{160}$ a reduced t_R compared to the lower resolution simulations was found. This increased the local slope between $N = 10^{80}$ and $N = 10^{160}$ (dotted lines, hardly distinguishable from the dashed lines, with the slope indicated below the line). Altogether, Fig. 8 suggests that the agreement between the numerical simulation and the monotrophic model be-

comes better as N increases, a conjecture already made by Malyshkin & Goodman (2001).

3.4. Fragmentation

The grouping method can also be used in steady-state situations where fragmentation balances coagulation. From the simulations in § 3.1 we expect that for the distribution method this is not a problem, as it follows both the number as well as the mass density. We will therefore concentrate our efforts on the equal-mass method. Fragmentation events in Monte Carlo may seem problematic, because catastrophic destruction can result in the creation of many small particles that consume computational resources. However, due to the grouping method we now have a natural way to deal with fragmentation. In this section we consider a test case of complete destruction – meaning: the breakup of a particle into its smallest constituents (monomers) – that occurs once a mass threshold, m_{frag} , is exceeded. We then solve for the emergent, steady-state, mass distribution. Note that due to stochastic fluctuations the number distribution is never really steady-state. We will therefore average over many distributions during the same simulation run to obtain the ‘steady-state’ mass spectrum.

In Fig. 9 the results of the sum kernel, $K_{ij} = m_i + m_j$, are given. In Fig. 9a the fragmentation threshold is $m_{\text{frag}} = 10^{10}$. The thick crosses denote the average over ~ 100 distributions during the simulation after the initial distribution has relaxed to semi-steady state. The volume of the simulation is stabilized when it incorporates $N_s^* = 10^4$ species at $\mathcal{V} = 2 \times 10^{12} \gg m_{\text{frag}}$. Figure 9b presents the result for an even larger value of $m_{\text{frag}} = 10^{20}$. The same characteristic distribution – resembling a bath tub – is found. These results suggest that, for the sum kernel with catastrophic fragmentation at a cut-off mass, most of the mass density becomes located near the end points. The system may then be approximated by two states at $m = m_0 \sim 1$ and $m = m_f \sim m_{\text{frag}}$. The mass-loss of the monomer state due to collisions with the high- m state is $mdm/dt|^- = n_0 n_f m_f$ and the gain due to fragmentation within the high mass bin becomes $mdm/dt|+ = n_f^2 m_f^2$. Balancing these we get $n_0 = n_f m_f$, i.e., equal mass densities for the monomer and high- m state.

However, with other kernels the mass distribution due to fragmentation may also emerge as a power-law. To see this we adopt a simple model (Camacho 2001) in which the mass sweep-up, \dot{m} , of a particle of mass m due to particles of lower mass $m' < m$ is assumed to change only the mass in a continuous way. That is,

$$\dot{m} = \int_{m_0}^m dm' K(m, m') m' f(m'), \quad (21)$$

will shift particles of mass m by an amount $\Delta m = \dot{m} \Delta t$. The total particle change per unit time in the interval $[m, m + \Delta m]$ becomes $-\Delta[\dot{m} f(m)]$ due to gradients in \dot{m} and the distribution function. The other side of the distribution then acts as a sink to the particles of mass m , i.e., $R(m) f(m) \Delta m$ is the number of particles in $[m, m + \Delta m]$ that is ‘consumed’ by the more massive particles, with $R(m)$

$$R(m) = \int_m^{m_{\text{frag}}} dm' K(m, m') f(m'), \quad (22)$$

the removal rate. For a steady state we must therefore have

$$\frac{\partial(\dot{m} f(m))}{\partial m} = -f(m) R(m). \quad (23)$$

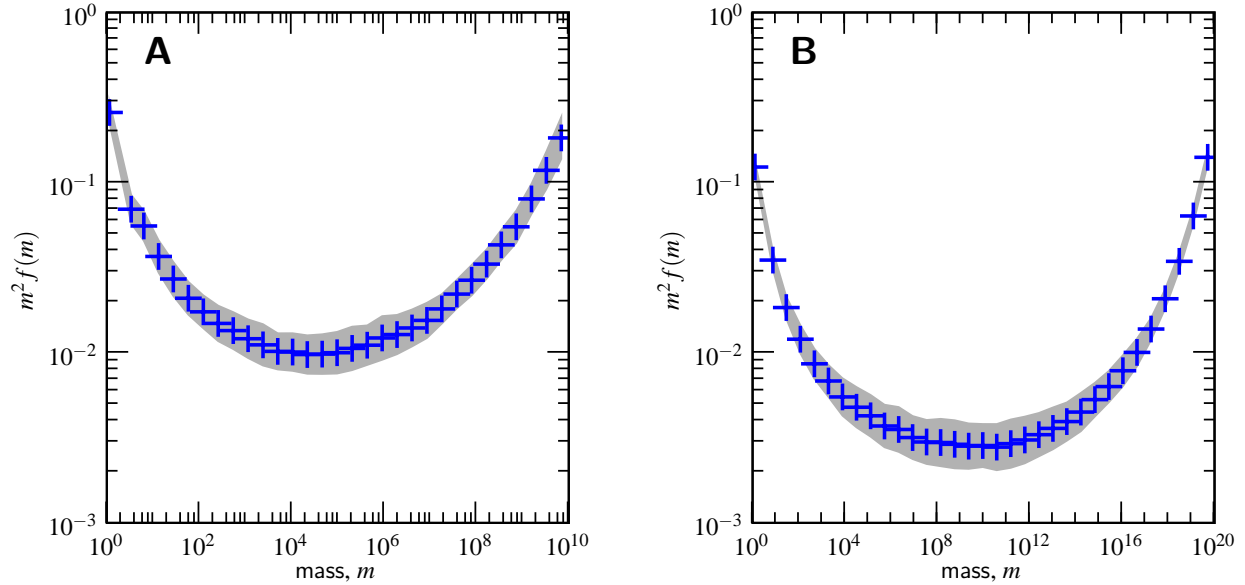


FIG. 9.— Steady state distributions of the sum kernel. At $m = m_{\text{frag}}$ particles are catastrophically disrupted into monomers of mass $m = m_0 = 1$. (A) $m_{\text{frag}} = 10^{10}$ and $N_s^* = 10^4$; (B) $m_{\text{frag}} = 10^{20}$ and $N_s^* = 2 \times 10^4$. Distributions are averaged over various times during the steady-state process. Crosses denote this average over ~ 100 distributions. The spread is indicated by the grey shading.

Assuming a power-law for f , $f \propto m^\alpha$, equation (23) can be solved for α by extending the limits of the integrals in equation (21) and equation (22) to 0 and ∞ , respectively (Camacho 2001). If $|\beta| < 1$ the integrals converge and the resulting power-law exponent is $\alpha = -(3 + \beta)/2$. The physical reason for the scale-free solution is thus that, contrary to the linear kernel, the behavior of the distribution near the cut-off points is unimportant for the (local) distribution. Other works have confirmed this power-law exponent, (e.g., White 1982; Hayakawa 1987). If $\beta > 1$ the integrals do not converge: i.e., collisions with the boundaries of the distribution dominate and no power-law emerges. However, Klett (1975) has shown that the $\beta = 1$ kernel $K_{ij} = (m_i m_j)^{1/2}$, which therefore somewhat resembles the sum kernel, has a $\alpha = -2$ power-law.

To test whether a power-law emerges, we have performed fragmentation simulations for the constant kernel, $K = 1$, see Fig. 10. Thus, as $\beta = 0$ we expect to end up with a slope of $1/2$ for the $m^2 f(m)$ mass distribution. This trend is indeed observed in Fig. 10a but only at large masses. The reason why an extension of the power-law to low masses is not seen, is the same as in Fig. 5a. Again, due to the equal-mass method, region of low mass density are not represented. Additionally, the injection rate at $m = m_0$ is episodic: at one instance many monomers of total mass $\sim m_{\text{frag}}$ enter the system, such that the mass density at this particular instance becomes very high (in comparison to the expected power-law distribution) for a very brief period of time. However, as coagulation is a non-linear process, i.e., $dn_0/dt \propto -n_0^2$, these particles coagulate very quickly with each other. In other words, the episodic injection rate has the effect of speeding up the coagulation compared to a smooth rate and, when averaged over time, the true distribution is underestimated at places where the particle density strongly fluctuates with time.

However, this deficiency of the equal-mass method does not affect the results at higher m : here the distribution is well resolved. Additionally, the mean (time-averaged) influx rate of monomers, I , at $m = m_0$ is available, as well as the removal rate of the higher mass particles, $R(m)$. A simple remedy to retrieve the low- m distribution becomes then available. In a separate simulation we have focused only on the low- m part

of the mass spectrum until a cut-off mass m_{cut} , and additionally include the fixed rates I and R , which were obtained from the previous simulation. Thus, $R(m)$ is the probability that a particle of mass m collides with *any* particle of the ‘frozen’ large- m distribution that was previously obtained in Fig. 10a. Finally, if particles manage to grow above m_{cut} (i.e., avoid to collide with a high- m particle through R) they are also removed. In this way we effectively re-perform the same fragmentation simulation, but in a reduced simulation volume, where the much smaller inter-collision time-increments Δt are appropriate for the low- m tail. In Fig. 10b this procedure is illustrated. Here, we put the cut at $m \approx 10^5$, recalculated the steady-state for small m , and ‘glued’ it to the high- m distribution of $m > m_{\text{cut}}$ from Fig. 10a.

4. SUMMARY AND DISCUSSION

4.1. Merits of the Method

In the previous section we have applied the grouping method to a variety of situations and found satisfactory behavior, both in terms of accuracy as in required computing power, despite the large dynamic range under consideration. The collisional evolutions of the analytic kernels were tested and agreed with the theoretical distributions. We also confirmed the predictions made for a class of runaway kernels, which required us to model an unprecedented large number of particles. Finally, we applied the grouping method to situations in which fragmentation operates and found excellent agreement with earlier findings (at least for the constant kernel; we are not aware of a prediction for the shape of the distribution function in the sum-fragmentation kernel).

Here we further discuss the (dis)advantages of both the equal mass and distribution methods and compare them with other studies. In short our conclusions are the following: the equal mass method is a true MC method, in the sense that no merging of the species array is required, while the mass density of the distribution is resolved (following a characteristic mass). For non-runaway situations this is therefore the preferred method. Other work along these lines can be found in Shima et al. (2007). These authors implement a superparticle method to model rain drop formation. These super-droplets

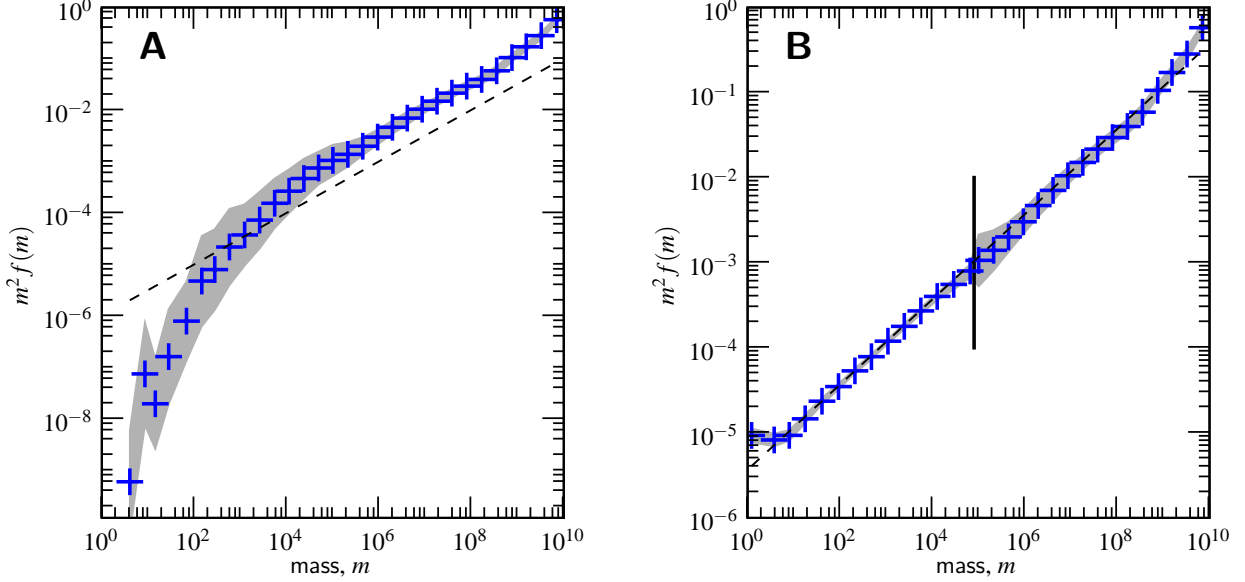


FIG. 10.— (A) Catastrophic fragmentation in the constant kernel, $K = 1$, with $m_{\text{frag}} = 10^{10}$. At masses above $m \sim 10^4$ a power-law develops with a slope approximately equal to the predicted slope ($\alpha = -3/2$; dashed line). At low- m the simulation lacks resolution and the power-law relation breaks down because of the episodic nature of the monomer injection. (B) This discrepancy is solved by performing a follow-up simulation for the low-mass part of the distribution only ($m < m_{\text{cut}} = 10^5$) but including a removal rate due to collisions with large masses above m_{cut} , $R(m)$, and a monomer injection rate, I , that were obtained in (A). The distribution above m_{cut} is copied from (A).

(like our groups) represent a multiple number of physical particles with the same properties (like mass) and position. However, their collisional setup is different. Where we allow many particles of one group to collide with another, in Shima et al. (2007) there is at most one collision per particle. Also, in the Shima et al. (2007) work the number of superdroplets is conserved and therefore (if we understand their picture correctly) the mean mass of the superdroplets. Thus, they always need a large number of superdroplets in order to resolve the high- m fluctuations (see the discussion on Fig. 2c of Shima et al. 2007). The biggest difference, therefore, is that our method allows a dynamical adjustment of the group sizes. Thus, we think the Shima et al. (2007) work is not capable of achieving the high dynamic range of our grouping method. Still, we do note that their method achieves growth factors of at least 10^6 in mass, and this may be sufficient for application to the atmospheric conditions of rain formation.

With the distribution method we are able to trace well the fluctuations of the mass (or distribution) spectrum, by choosing the zoom factors such that there are a fixed number of groups for all exponentially spaced mass bins. This method is therefore well suited for runaway kernels. However, the consequence is a rapid proliferation of the N_s parameter as many new species (which may be insignificant by, e.g., mass) are created. To limit N_s , these species need to be merged, which somewhat violates the spirit of MC-methods as this includes averaging, not only over mass but also over structural parameters. The merging feature renders this method similar to Wetherill's discrete 'batches' technique (Wetherill 1990). However, in our method structural parameters, like porosity and charge, can be included and there is considerable (useful) freedom in how the averaging can be performed. These structural parameters survive, in averaged form, the merging process in our distribution method, and can be fundamental in the process of coagulation. For example, fractal grain structure (porosity) and dust charging can lead to, respectively, strong particle growth and even gelation (Konopka et al. 2005; Ormel et al. 2007).

Both the Wetherill (1990) method and our approach give the correct behavior for the most massive and second most massive particles in the case of the product kernel (see Inaba et al. 1999 for verification). Still, Wetherill's method has to deal with a number of free parameters: the spacing between the discrete batches and a parameter that determines the timestep, which must be fine-tuned. Our MC-method, although not free of them either, does have the advantage that the free parameters correspond to physically-identifiable quantities, such as the mass-fraction parameter, f_e . Another key advantage of our code is that the timestep, Δt , follows naturally from the MC-code and is not required to be specified a priori. Especially in the super-runaway kernels with $\nu = 2, 3$, growth can be very erratic: during the runaway-process there is a large amount of growth during a small time step. The success of Wetherill's approach lies in 'discretizing' the original (mean-field) Smoluchowski equation in an efficient way. Naturally, in Monte Carlo codes, discreteness is assured.

The existence of two separate methods might appear to be a drawback. That is, runaway kernels (modeled with the distribution method) and information on structural parameters (through the equal mass method) are difficult to treat simultaneously, especially because one does not know a priori if a runaway situation will materialize. However, information about the kernel is 'hard-wired' into the Monte Carlo program and it is therefore clear at which point the situation is susceptible to runaway growth. Then, these particles (the high- m particles) have to be further resolved, and this can be achieved by switching to the distribution method. Thus, we feel that it is possible to model runaway kernels with MC-methods, provided one implements a hybrid switch that signals runaway growth through the effective mass scaling exponent that a kernel exhibits during the simulation. For example, consider the cross section of equation (5), the mass scaling of σ , m^ν , becomes stronger than unity ($\nu \approx 4/3$) for sufficiently large masses. It does so in a smooth manner, allowing a hybrid switch to be implemented. That is, as the program signals a large value of ν for a particular kernel, it switches

from the equal mass to the distribution method, i.e., assigning lower zoom factors for the high- m particles and, if needed, merging structural parameters. This kernel adaptation technique is analogous to adaptive mesh refinement (Wise & Abel 2007), where the latter is grid based/spatial and we take a particle/stochastic approach. Both numerical treatments require one to look carefully at the consequences of the switch, i.e., the impact of the merging prescription or the under-resolved part of the grid.

The bottom-line on MC-methods thus becomes clear: there is *a priori* no universal way in which the correct, accurate result can be guaranteed. Each situation requires interpretation and, if necessary, (re)adjustment of some aspects related to the grouping method; most notably through the choices for the grouping factors, i.e., the $\{z_i\}$. However, MC-methods are flexible enough to do so in physically meaningful way. As such, they are a very powerful tool to describe any dynamical system in which particles interact in a manner that can be statistically represented. In fact, the techniques presented in this work allow one to treat a wide range of, astrophysically relevant, kernels. The nicest feature of MC-methods is that besides mass, structural properties of particles, like porosity and charge, can be directly incorporated into the collisional dynamics.

4.2. Astrophysical implications

There are several astrophysical situations where a large dynamic range is required and for which our algorithm may be applicable. In protoplanetary disks, one of the key questions is how the tiny (sub)micron-sized grains are transferred to planetesimals (Dominik et al. 2007). Because of the increasing relative velocities with particle growth, it may very well be the case that, while grains grow, the likelihood of fragmentation increases. An additional argument for fragmentation is that collisional growth very efficiently removes the smallest grains, contrary to what observations indicate (Dullemond & Dominik 2005). Recently, Brauer et al. (2007) has solved the steady-state coagulation-fragmentation equation for the dust distribution. It would be worthwhile to investigate whether these results are also obtained with the grouping method, and how this picture changes by inclusion of structural parameters, e.g., the porosity of aggregates (Ossenkopf 1993; Kempf et al. 1999; Ormel et al. 2007).

If particles reach the planetesimal size (~ 1 km) self-gravity may be sufficient to keep the collision products together. Fur-

thermore, gravitational focussing, enhancing the gravitational cross-section, will turn the accretion process into a runaway process ($\nu > 1$). Goldreich et al. (2004) has outlined under which conditions runaway growth proceeds. Greenberg et al. (1978) followed the collisional evolution (including fragmentation) of a population of initially 10^{12} km-sized planetesimals until sizes of ~ 500 km were reached (after which particle-in-a-box approximations fail). We again note that a bin-based approach may have difficulty to model the runaway stage unless it uses an ad-hoc extension to include the runaway particle explicitly (Wetherill 1990). Bromley & Kenyon (2006) have developed a hybrid code that combines a N-body and statistical approach. However, these models can only treat one variable; with a Monte Carlo approach the internal structure of the particles can be included and a rubble pile structure can be distinguished from a homogeneous one.

In stellar clusters, too, conditions may be feasible for runaway growth. Due to dynamical friction (equipartition) the more massive stars will sink to the center of the cluster and we may even have $\nu = 2$ (Mal'ushkin & Goodman 2001; Lee 2000). Moreover, the cluster will eject the smaller members and contract as a whole. These processes lead to *core collapse* in which the inner cut-off for the power law density function, r_0 , disappears, $r_0 \rightarrow 0$, and the density seemingly becomes singular. This condition is very favourable for a runaway-growth scenario, and may well be the mechanism for the formation of high mass stars, seed black holes or even gamma ray bursts (Portegies Zwart et al. 1999; Freitag et al. 2006; Portegies Zwart & van den Heuvel 2007). Globular clusters are already successfully modeled by MC-methods where the structural parameters are now the energy and angular momentum characterizing an orbit on which a particle (group) resides (e.g. Joshi et al. 2000). Here, the grouping mechanism may then be well suited to model systems with a very large number of stars and stellar ejecta. Similarly, the collapse, and possible fragmentation, of an interstellar gas cloud, leading to the formation of a stellar cluster or black hole, could be treated by following many (porous) gas clumps.

We appreciate the efforts of Volker Ossenkopf to proof-read the manuscript and the useful suggestions and comments of the two referees, which significantly improved the structure and presentation of this work.

REFERENCES

- Blum, J. 2004, in ASP Conf. Ser. 309: Astrophysics of Dust, ed. A. N. Witt, G. C. Clayton, & B. T. Draine, 369
- Bonabeau, E., Dagorn, L., & Freon, P. 1999, Proceedings of the National Academy of Science, 96, 4472
- Brauer, F., Dullemond, C. P., & Henning, T. 2007, ArXiv e-prints, 711
- Bromley, B. C., & Kenyon, S. J. 2006, AJ, 131, 2737
- Camacho, J. 2001, Physical Review E, 63, 046112
- Camacho, J., & Solé, R. V. 2001, Europhysics Letters, 55, 774
- Dominik, C., Blum, J., Cuzzi, J. N., & Wurm, G. 2007, Protostars and Planets V, 783
- Dullemond, C. P., & Dominik, C. 2005, A&A, 434, 971
- Freitag, M., Rasio, F. A., & Baumgardt, H. 2006, MNRAS, 368, 121
- Gillespie, D. T. 1975, Journal of Atmospheric Sciences, 32, 1977
- Goldreich, P., Lithwick, Y., & Sari, R. 2004, ARA&A, 42, 549
- Greenberg, R., Hartmann, W. K., Chapman, C. R., & Wacker, J. F. 1978, Icarus, 35, 1
- Hayakawa, H. 1987, Journal of Physics A Mathematical General, 20, L801
- Inaba, S., Tanaka, H., Ohtsuki, K., & Nakazawa, K. 1999, Earth, Planets, and Space, 51, 205
- Jeon, I. 1998, Communications in Mathematical Physics, 194, 541
- Joshi, K. J., Rasio, F. A., & Portegies Zwart, S. 2000, ApJ, 540, 969
- Kempf, S., Pfalzner, S., & Henning, T. K. 1999, Icarus, 141, 388
- Klett, J. D. 1975, Journal of Atmospheric Sciences, 32, 380
- Konopka, U., Mokler, F., Ivlev, A. V., Kretschmer, M., Morfill, G. E., Thomas, H. M., Rothermel, H., Fortov, V. E., Lipaev, A. M., Molotkov, V. I., Nefedov, A. P., Baturin, Y. M., Budarin, Y., Ivanov, A. I., & Roth, M. 2005, New Journal of Physics, 7, 227
- Kovetz, A., & Olund, B. 1969, Journal of Atmospheric Sciences, 26, 1060
- Laurenzi, I. J., Bartels, J. D., & Diamond, S. L. 2002, Journal of Computational Physics, 177, 418
- Laurenzi, I. J., & Diamond, S. L. 2003, Phys. Rev. E, 67, 051103
- Lee, M. H. 2000, Icarus, 143, 74
- . 2001, Journal of Physics A Mathematical General, 34, 10219
- Leyvraz, F. 2003, Phys. Rep., 383, 95
- Lissauer, J. J. 1993, ARA&A, 31, 129
- Mal'ushkin, L., & Goodman, J. 2001, Icarus, 150, 314
- Marshall, J., & Cuzzi, J. 2001, in Lunar and Planetary Inst. Technical Report, Vol. 32, Lunar and Planetary Institute Conference Abstracts, 1262+—
- Ormel, C. W., & Cuzzi, J. N. 2007, A&A, 466, 413
- Ormel, C. W., Spaans, M., & Tielens, A. G. G. M. 2007, A&A, 461, 215

- Ossenkopf, V. 1993, A&A, 280, 617
 Portegies Zwart, S. F., Makino, J., McMillan, S. L. W., & Hut, P. 1999, A&A, 348, 117
 Portegies Zwart, S. F., & van den Heuvel, E. P. J. 2007, Nature, 450, 388
 Shima, S.-i., Kusano, K., Kawano, A., Sugiyama, T., & Kawahara, S. 2007, ArXiv Physics e-prints
 Silk, J., & Takahashi, T. 1979, ApJ, 229, 242
 Smith, M., & Matsoukas, T. 1998, Chem. Eng. Sci., 53, 1777
 Smoluchowski, M. V. 1916, Zeitschrift fur Physik, 17, 557
 Spaute, D., Weidenschilling, S. J., Davis, D. R., & Marzari, F. 1991, Icarus, 92, 147
 Spouge, J. L. 1985, J. Colloid Interface Sci, 107, 38
 Tanaka, H., & Nakazawa, K. 1994, Icarus, 107, 404
 Trubnnikov, B. A. 1971, Doklady Akad. Nauk USSR, 196, 1316
 Wetherill, G. W. 1990, Icarus, 88, 336
 White, W. H. 1982, J. Colloid Interface Sci, 87, 204
 Wise, J. H., & Abel, T. 2007, ApJ, 665, 899

APPENDIX

GROUPED COLLISION RATES

We justify the expressions for the grouped collision rates, equation (10), where the collision rates are divided by the number of collisions during the group collisions. We show that, provided the individual rates C_{ij} do not change as particles accrete, the mean collision rate is independent of the grouping.

Consider a species of N identical particles that react with an individual rate a_0 with an external particle. In the language of the grouping method of § 2 the N particles are of species j that, during the group process, react with a particle from species i and $a_0 = C_{ij}$. Let the process be such that the N particles react in groups of N/w , where $1 \leq w \leq N$ is the total number of such events.

Let P_i be the probability that the system is in state i , meaning that the i -th event (out of w) has occurred. At $t = 0$ the system is in the first state, so $P_0(t = 0) = 1$ and $P_i(t = 0) = 0$ for $i \neq 1$. Let the rate at which the system evolves from state i to the next state $(i + 1)$ be given by λ_i , then

$$\frac{dP_0}{dt} = -\lambda_0 P_0; \quad (A1)$$

$$\frac{dP_1}{dt} = -\lambda_1 P_1 + \lambda_0 P_0; \quad (A2)$$

$$\frac{dP_i}{dt} = -\lambda_i P_i + \lambda_{i-1} P_{i-1}; \quad (1 < i < w) \quad (A3)$$

$$\frac{dP_w}{dt} = +\lambda_{w-1} P_{w-1}. \quad (A4)$$

Let $\overline{R}(t)$ be the mean reactivity of the system measuring the mean number of particles that has reacted at time t ($0 \leq R \leq 1$). Likewise, let $\overline{R^2}$ measure the square number of reactants, i.e.,

$$\overline{R^k}(t) = \frac{1}{N^k} \sum_{i=0}^w P_i \left(\frac{iN}{w} \right)^k = \sum_{i=0}^w P_i \left(\frac{i}{w} \right)^k. \quad (A5)$$

If we weigh the state-equations (eq. [A4]) with i/n and $(i/n)^2$, respectively, and sum over i we obtain:

$$\begin{aligned} \frac{d\overline{R}}{dt} &= \frac{1}{w} \sum_{i=0}^w \lambda_i P_i; \\ \frac{d\overline{R^2}}{dt} &= \frac{1}{w^2} \sum_{i=0}^w (2i + 1) \lambda_i P_i. \end{aligned} \quad (A6)$$

Now the λ_i are proportional to the amount of particles left, i.e., $\lambda_i \propto (1 - i/w)N$. In the case of group collisions, furthermore, we have argued that the rates should be divided by the number of reactants, such that the collision rate stays the same. We thus obtain

$$\lambda_i = (w - i)a_0, \quad (A7)$$

where a_0 is the individual collision rate between two single particles. Equation (A6) then transform to

$$\frac{d\overline{R}}{dt} = a_0 (1 - \overline{R}); \quad (A8)$$

$$\frac{d\overline{R^2}}{dt} = a_0 \left(-2\overline{R^2} + \frac{2w-1}{w} \overline{R} + \frac{1}{w} \right). \quad (A9)$$

This first equation of the first moment (average value) is independent of w . Its solution is of course the well known exponential decay corresponding to Poisson statistics, i.e.,

$$\overline{R}(t) = 1 - \exp[-at]. \quad (A10)$$

Inserting \overline{R} into the equation for the second moment we solve for $\overline{R^2}$ as

$$\overline{R^2}(t) = (1 - e^{-at}) \left(1 - e^{-at} (1 - w^{-1}) \right). \quad (A11)$$

The spread σ in the distribution then becomes

$$\sigma^2(w) = \left(1 - e^{-at}\right) \frac{e^{-at}}{w}. \quad (\text{A12})$$

That is, for $w \rightarrow \infty$, $\sigma \rightarrow 0$ and we are certain about the fraction of objects that have collided. For small w , however, there can be a significant spread.

By choosing the collision rates as in equation (A7) – which is simply the rate for a collision with one particle, $a_0(N - iN/w)$, divided by the number of particles in the group (N/w) – we end up with equation (A10) for the mean number of collisions. Since this is independent of w the mean reactivity of the system is also unaffected by the choice of w . For the mean reactivity it does not matter whether $w = N$ (no grouping) or $w = 1$ (all particles in the same group) as long as a_0 stays constant during the group process. However, the spread in \bar{R} does depend on the choice of w , which is quite natural, of course. A similar procedure cannot be performed for the in-group collisions since here equation (A4) involves i^2 terms. The group collision rates for in-group collisions are not exact. However, the rates of equation (10) are still a good approximation, and will become exact for large w . Besides, in-group collisions are a relatively minor occurrence.